DOCUMENT RESUME

ED 152 214 HE 009 811

AUTHOR	Hussain, K. M., Ed.
TITLE	Management Information Systems for Higher Education. Studies in Institutional Management in Higher Education.
INSTITUTION	Organisation for Economic Cooperation and Development, Paris (France). Centre for Educational Research and Innovation.
PUB DATE	77
NOTE	346p.; Some pages may not reproduce well due to marginal legibility of original document
AVAILABLE FROM	Director of Information, Organisation for Economic Development and Cooperation, 2, rue Andre-Pascal, 75775 Paris Cedex 16, France ($13.50)
EDRS PRICE	MF-$0.83 Plus Postage. HC Not Available from EDRS.
DESCRIPTORS	Annotated Bibliographies; Church Related Colleges; College Administration; Curriculum Planning; Foreign Countries; Higher Education; Library Technical Processes; *Management Information Systems; *Management Systems; Quality Control; *Systems Analysis; *Systems Concepts; *University Administration
IDENTIFIERS	Bath University of Technology (England); Catholic University of Leuven (Belgium); Laval University (Quebec); *New Colleges; New University of Lisbon (Portugal)

ABSTRACT

Contents include an introduction to information systems (basic concepts and life-cycle); development of an information system (design, implementation, testing, and conversion); operation (evaluation and maintenance, quality control, and economics); and case studies of such systems at the New University of Lisbon, Bath University, Laval University (course management), and the Catholic University of Leuven. An annotated bibliography is included. (MSE)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED152214

Centre for Educational Research and Innovation (CERI)

STUDIES IN INSTITUTIONAL MANAGEMENT
IN HIGHER EDUCATION

MANAGEMENT INFORMATION SYSTEMS FOR HIGHER EDUCATION

Edited by
K.M. Hussain, Ph. D.
Professor
New Mexico State University
USA

based on presentations made at
a Professional Seminar
organised by CERI's Programme
on Institutional Management
in Higher Education
in June 1975.

HE009811

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

OECD

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC) AND USERS OF THE ERIC SYSTEM."

ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT
1977

2

The Organisation for Economic Co-operation and Development (OECD) was set up under a Convention signed in Paris on 14th December, 1960, which provides that the OECD shall promote policies designed:

- — to achieve the highest sustainable economic growth and employment and a rising standard of living in Member countries, while maintaining financial stability, and thus to contribute to the development of the world economy;
- — to contribute to sound economic expansion in Member as well as non-member countries in the process of economic development;
- — to contribute to the expansion of world trade on a multilateral, non-discriminatory basis in accordance with international obligations.

The Members of OECD are Australia, Austria, Belgium, Canada, Denmark, Finland, France, the Federal Republic of Germany, Greece, Iceland, Ireland, Italy, Japan, Luxembourg, the Netherlands, New Zealand, Norway, Portugal, Spain, Sweden, Switzerland, Turkey, the United Kingdom and the United States.

The Centre for Educational Research and Innovation was created in June 1968 by the Council of the Organisation for Economic Co-operation and Development for an initial period of three years, with the help of grants from the Ford Foundation and the Royal Dutch Shell Group of Companies. In May 1971, the Council decided that the Centre should continue its work for a period of five years as from 1st January, 1972.

The main objectives of the Centre are as follows:

- *— to promote and support the development of research activities in education and undertake such research activities where appropriate;*
- *— to promote and support pilot experiments with a view to introducing and testing innovations in the educational system;*
- *— to promote the development of co-operation between Member countries in the field of educational research and innovation.*

The Centre functions within the Organisation for Economic Co-operation and Development in accordance with the decisions of the Council of the Organisation, under the authority of the Secretary-General. It is supervised by a Governing Board composed of one national expert in its field of competence from each of the countries participating in its programme of work.

⁂

© OECD, 1977
Queries concerning permissions or translation rights should be addressed to:

Director of Information, OECD
2, rue André-Pascal, 75775 PARIS CEDEX 16, France.

CONTENTS

Preface 5

Chapter 1

Introduction 7

Part One

INTRODUCTION TO INFORMATION SYSTEMS

Chapter 2

Information Systems: Basic Concepts 13

Chapter 3

Life Cycle of an Information System 29

Part Two

DEVELOPMENT OF AN INFORMATION SYSTEM

Chapter 4

Data Base: Design and Implementation 45

Chapter 5

Design of an Information System 103

Chapter 6

Testing and Conversion of an Information System 121

Part Three

OPERATING AN INFORMATION SYSTEM

Chapter 7

Evaluation and Maintenance of an Information System . . 143

Chapter 8

Quality Control of Information 151

Chapter 9

Organizing Information Systems 171

Chapter 10

Economics of an Information System 193

Part Four

CASE STUDIES ON INFORMATION SYSTEMS IMPLEMENTED
IN IMHE INSTITUTIONS OUTSIDE THE U.S.

Chapter 11

Developing a Data Base for a New Educational Institution: A case study of Universidade Nova de Lisboa 215

Chapter 12

Automated Issue System - Bath University Library 237

Chapter 13

Laval System for Course Management 247

Chapter 14

Towards an Integrated Management Information System at the Katholieke Universiteit te Leuven 283

Selected Annotated Bibliography (by chapter) 316

Glossary 325

Preface

Over the past few years, a considerable amount of effort has been expended to devise better ways of resolving administrative and planning problems faced by institutions of higher education. As a result of research, and experience, a certain amount of basic knowledge and management expertise has been accumulated in a number of problem areas. The subject of this book is limited to one such area, namely the design and implementation of information systems for institutions of higher education. It is an outgrowth of one in a series of professional training seminars organised by OECD's Programme on Institutional Management in Higher Education to introduce university personnel with management responsibility to some basic concepts and approaches to university management.

Information has become increasingly important for management, including that of higher education. It is needed for operations, control and planning and at all levels: institutional, regional and national. For getting this information, we need to design and implement information systems, a subject that is becoming complex and costly with the advent of computers.

Because of the cost and importance of the system it is necessary that managers and administrators in higher education be knowledgable about information systems. It is for providing such knowledge that the OECD's Programme on Institutional Management in Higher Education organised a seminar in June 1975 consisting of a set of lectures and case studies, some of which have been selected for inclusion in this volume. The main criterion of selection has been the relevance to higher education management and the unavailability of the material in the literature on higher education as well as in the literature on business information systems.

The papers and cases have been revised and edited. The purpose is to provide the manager with some knowledge necessary so that he can communicate with his technical personnel and make decisions related to information processing.

The roles of a number of organisations and individuals in the preparation of the publication should be acknowledged. Firstly, it should be noted that the professional training seminars were made possible as a result of a grant of $100,000 from the Ford Foundation covering a two-year period. Secondly, there are many individuals and organisations who made important contributions: Ann Chapman, James Farmer, the Katholieke Universiteit te Leuven, J.H. Lamble, Marcel Lauzon and Charles Thomas. Finally, K.M. Hussain contributed many papers included in this volume and did the editing. All these contributions are greatly appreciated.

J.R. GASS
Director
Centre for Educational
Research and Innovation

Chapter 1

INTRODUCTION

In the past decade there has been an increasing demand for the services provided by information systems. The world of microcircuitry and transmission of data at speeds approaching that of light is generating information at an explosive rate. Administrators of organizations in this environment expect information that is precise, accurate and timely. This information is used for record-keeping as well as for decision-making, using techniques such as operations research, econometrics, systems analysis, and PPBS (Program, Planning, and Budgeting Systems). Information has been used successfully in business and industry for these purposes in situations similar to those arising in educational institutions. Consequently, it is postulated that the performances of educational institutions could be upgraded by the use of properly designed and implemented information systems.

In industry there is evidence that information systems contribute to higher profits (resulting from efficiency) and greater effectiveness. Thus, there is a demand from legislatures that such systems also be used in education, where the growth rate is high and is continuing to increase.

There are also non-economic forces that increase the demand for information systems in education; for instance, the demands for more information on educational operations from state, regional, and national agencies. In addition to external demands, there are internal pressures. Students are demanding a faster and more effective system for handling their records. Since they can inquire about the availability of an airplane seat and make instant reservations on the phone, they expect equally fast and effective responses at the time of their registration for classes. Faculty members are making similar demands for more and faster information, which is necessary for their work to be effective. All these pressures increase the need for information systems in education.

Information is power. It can be used, or it can be abused. Also information can be good - accurate, timely and relevant - or it can be bad - inaccurate, late and irrelevant. But information systems can be developed to generate information that is good and useful to management in higher education.

The development of such systems is largely a technical task, one that requires system analysis, computer programmers and other computer specialists. These technicians cannot start their work without the needs of the system being specified by management. This will require detailed definitions of terms, concepts, and decision rules, the specification of variables and information and their limits of acceptance and validity as well as specification of desired output and available input. Finally, it requires the specification of the testing of the system and its implementation.

All these activities are the responsibility of management - sometimes top management, sometimes supervisory management and sometimes operations clerks. They must not delegate their responsibility to the technician who often does not have the necessary background perception, or additional knowledge to perform this role. Sometimes the technician performs the role but often by default because the manager was too busy or had little interest in performing his function. This results in a system that is ineffective. To prevent this, management must participate in the development of its information system and perform the role that it alone can perform.

To perform its responsibilities in the development of an information system, management must have a basic understanding of the process of the development of an information system. To provide such an understanding, the Centre for Educational Research and Innovation at the OECD offered a seminar(1) for management in higher education, at which numerous papers were presented. The evaluation by the participants showed a desire for the papers to be edited and published. This demand has resulted in this book.

Organization of book

This book has four parts: Part One covers basic concepts of an information system (Chapter 2) as well as an overview of the development of an information system (Chapter 3). Some selected stages of the development process are discussed in Part Two. These are: the development of a data base (Chapter 4); the design of an information system (Chapter 5) and the testing and conversion of an information system (Chapter 6). Once the system is ready for conversion, it is operated. This is discussed in Part Three. In it are discussed the evaluation and maintenance of the information system (Chapter 7); the control of the quality of information (Chapter 8); the organization of an information system (Chapter 9); and finally, the economics of an information system (Chapter 10). Throughout this discussion, the target audience is the manager in higher education.

---

1) Information Systems for Institutions of Higher Education, June, 1975.

Hence, only material which is important and relevant to the manager in higher education is discussed. This material is necessary to enable the manager to communicate with the technician and ensure that he gets the information system that meets his needs. Part Four is a set of case studies of information systems which have been implemented. These were recently implemented and selected to represent a sample of systems from the very simple to the rather complex; a series of different approaches to information system design; and finally a mix of systems in different types of institutions (new and mature) developed in different parts of the OECD community.

Outline of book

There is a brief introduction to each Part, identifying its contents and its relationship to the other Parts of the book. There is also an introduction to most chapters, followed by the text, a summary and conclusions, and a set of references. For the reader who wishes to pursue the subject further, there is an extensive annotated bibliography at the end of the book organized by chapter.

The introduction and the summaries could be read in a sequence and are designed to provide a summary of the book.

Prerequisites for book

There are no prerequisites in mathematics or computer science expected of the reader. All concepts are developed and all terms are defined. A glossary in prose is given at the end of the book. It gives a set of informal and operational definitions of terms related to information systems. This glossary describes the development, maintenance and operation of an information system. It can be read in any sequence - before the text, midway or at the end. It is, however, recommended that it be read at least at the beginning so that the reader and author establish a vocabulary.

Part One

INTRODUCTION TO INFORMATION SYSTEMS

This Part consists of two chapters. The first, Chapter 2, concerns basic definitions and concepts of an information system. The second, Chapter 3, is an overview of the stages and activities of the development of an information system. Each development has a life, with a start and an end and hence this chapter is entitled the life cycle of an information system. Some of these stages of development will be discussed in great detail in the next part of the book - Part Two.

Chapter 2

INFORMATION SYSTEMS:BASIC CONCEPTS(1)

by

K.M. Hussain

CONTENTS

1. Introduction 14
2. Systems 14
    2.1 Black-Box Concept 14
    2.2 Control and Feedback 16
3. Data and Information 18
4. An Information System 21
5. Subsystems 21
6. Integrated Systems and Subsystems 22
    6.1 Horizontal Integration 22
    6.2 Vertical Integration 23
    6.3 Longitudinal Integration 24
7. Total Systems and MIS 24
8. Summary and Conclusions 25

References 27

Annotated Bibliographies
(to be found at the end of the book)

A: On Basic Concepts of Information Systems 318
B: On Computers and Information Systems 321
C: On Systems Approaches to Educational Management . . . 323

---

1) This chapter is partly an extraction from Development of Information Systems for Education by Khateeb M. Hussain (c) 1973. It is reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, New Jersey, United States.

13

1. INTRODUCTION

In this chapter the basic concepts and components of an information system will be examined. What is a system? What is information? What is an information system?

A system has components called subsystems. The systems and subsystems that produce information in educational institutions will be discussed. To produce information that is consistent, accurate and economical, it is necessary to have the systems and subsystems integrated. The concept of integration and the many types of integration (horizontal, vertical and longitudinal) will be examined. When fully integrated, systems are sometimes called total systems or Management Information Systems (MIS). These concepts as well as their relevance and existence will also be discussed.

2. SYSTEMS

The term system is used in many contexts. In the natural sciences we speak of the solar system, molecular systems, and ecological systems. Discussing the human body, we can identify the circulatory system and the central nervous system. In daily life we encounter political systems, communications systems, highway systems, and many others. In all contexts, the word system denotes an assemblage of components or an order. One group of authors (Johnson et al., 1967: 113) defines a system as "an organized or complex whole, an assemblage or combination of things or parts forming a complex or unitary whole....it will be helpful to define systems more precisely as an array of components designed to accomplish a particular objective according to plan."

2.1 Black-box concept

A computer is part of a system and indeed an important part of many information systems. It can be thought of as a "black-box" with an input and output as shown in Figure 1.

There is input going into the computer and output coming out of it. Neither the mechanics nor the logic or the transformation from input into output are explained. The computer performing the transformation is unexplained and is treated as a black-box. We are not necessarily able to see into the box and understand its working but merely know the relationship of its input to its output.

Figure 1

THE BLACK-BOX APPROACH TO A COMPUTER

Input
Computer
Output

The black-box concept comes from the electrical engineer. He gets equipment in the form of a "box" which he may not open or examine; yet he needs to use it and hence to know how it behaves. He applies inputs in the form of voltages and currents, and measures the output on instruments. By experimentation with different inputs he is able to draw conclusions about the contents and behaviour of the box. The colour black of the box comes from the fact that it was once a popular colour for electrical equipment.

We use the black-box approach many times in our everyday lives. For example, we treat a plant as a black-box. We feed it water, sun, and fertilizer in order to produce growth but we do not necessarily know how the transformation takes place. We do not care to know, we do not have the ability to understand the technicalities, or do not find it always feasible to know. For example, it is not feasible to learn about the transformation relationship of a particular plant without digging it up and examining its root structure, but this may destroy the plant. We, therefore, are content with knowing the inputs (water, sun and fertilizer) required for a desired output (growth) and treat the plant itself as a black-box(1).

Many administrators (or managers) in higher education treat the computer as a black-box, much as they treat the engines under the hoods of their automobiles. To know the workings of internal combustion engines could help in the case of breakdowns and could otherwise contribute to more efficient operations and longer lives for the automobiles. But such knowledge of an automobile or a computer is not essential and they can be treated as black-boxes. What is necessary, however, is to understand the relationships of the inputs and outputs, their capabilities and limitations. Without this knowledge, the computer cannot be used effectively and, in fact, it may even be dangerous, just as an automobile can be dangerous if the driver does

---

1) For other examples of the "black-box" approach, see Van Court Hare, Jr., (1967) Systems Analysis: A Diagnostic Approach, New York: Harcourt, Brace and World, Inc., pp.20-37. For a more mathematical discussion, see Ashby, W.R. (1961). An Introduction to Cybernetics, London: Chapman and Hall, Ltd., pp.86-117. For a management approach, see Anshen, M., (1960), "The Manager and the Black-Box", Harvard Business Review, XXXVIII, No.6 (November-December), pp.85-92.

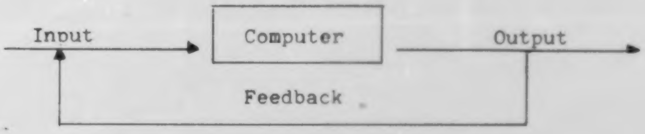
not know the relationship of the speed response to pressure on the accelerator. The administrator must know the relationship between inputs and outputs to a computer as well as to other man-machine combinations so that he may help design a system to meet his needs.

A system in this context is a grouping of men and machines to perform specific tasks. To design a system for his needs, the administrator must work closely with systems personnel. This group of people includes systems analysts, those who analyse and design systems; computer programmers, those who write computer programs; operations researchers, those who have knowledge of specialised mathematical and statistical methods and techniques used in systems; and other related professionals. The number of people in each profession working on a system development will vary with the nature and complexity of the system.

An important element in a systems operation is the "control" function: the correction of undesirable deviations of the actual performance from the desired performance. Information on this deviation is fed back to alter the input so that the undesirable deviations are eliminated. This feeding back of information is called feedback and is shown diagramatically in Figure 2.

Figure 2

DIAGRAM SHOWING FEEDBACK



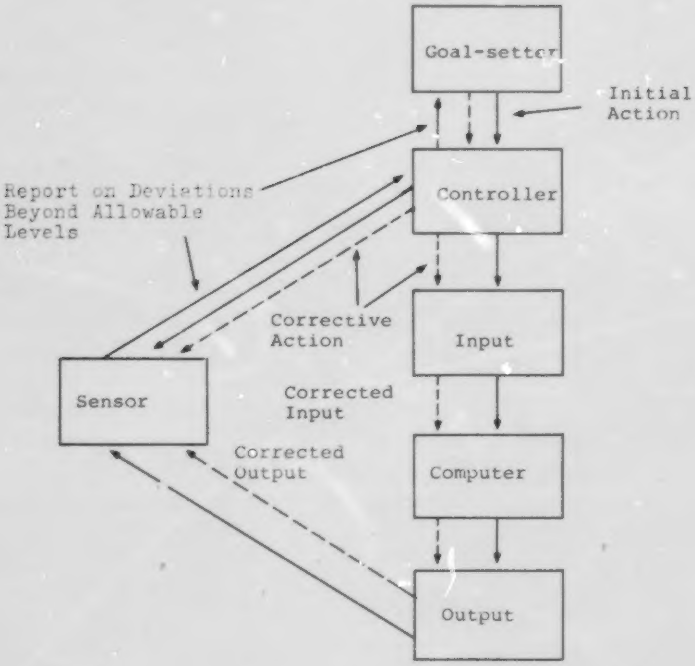
2.2 Control and feedback

The achievement of a single output at a point in time is not sufficient for a continuing system. The system must continue to perform according to the objectives and planned performance levels. Deviations beyond allowable tolerances must be detected (sensed) and corrected quickly and effectively. This can be done by a feedback mechanism, which is an essential part of the control function - the comparison of actual with desired performance and the initiation of appropriate corrective action. The feedback mechanism, when properly designed and used, will often anticipate undesirable deviations or trends, triggering corrective actions that will eliminate the causes of such deviations prior to the occurrence.

A simplified system with the control feature is illustrated in Figure 3. The initial actions are shown by the set of continuous

Figure 3

A SIMPLIFIED FEEDBACK SYSTEM


Goal-setter
Initial Action
Controller
Report on Deviations Beyond Allowable Levels
Corrective Action
Input
Sensor
Corrected Input
Corrected Output
Computer
Output

lines from goal-setter to output. The output is checked by a sensor which compares its deviations from the desired output against allowable deviations set by the controller. Deviations greater than those allowed are reported to the controller (middle manager), who then takes appropriate action modifying the input which in turn modifies the output. These modified actions are shown by the dashed lines. This cycle should be continuous during the life of the system. It enables the output to stay within specified limits decided upon by the controller, who in turn operates within the objectives and constraints specified by the top administrator (shown by continuous lines from goal-setter to controller and sensor).

Figure 3 identifies the important concepts that characterize systems: an assemblage of components forming a whole, designed for an objective and organised according to a plan. For an information system one can be more specific. In it, the output is information and the main input is data. The concepts of data and information will now be discussed.

3. DATA AND INFORMATION

Data consist of a set of characters or signals to which a significance can be assigned. Information, on the other hand, is selected data that have been processed to make them meaningful.

Consider a set of characters, 585281928F, stored in a space reserved for a student's social security number and sex code. These are data because the nine digits identify a person by his social security number, and the "F" is a code that identifies a person's sex. If the number is associated with a name, say Renée Schmal, and the F code identifies this person's sex, which when decoded means female, then we know that Renée Schmal is a female student. This is meaningful and is information. Thus the data, 585281928F, are information when processed (decoded and related to other data).

The importance of the "selection" of data can be illustrated by considering this problem further. If Renée Schmal were to register in a first year course, the data on sex would not constitute information to the registrar because sex is irrelevant when enrolling in a class and hence would not be "selected" for use. However, the data on sex would be selected by the Head Resident who has to assign Renée Schmal to a dormitory room on a coed campus. To the Head Resident, characters 585281928F (when decoded and related to other data) do constitute information. This selection aspect of information is emphasized by Schoderbek (1968: 44), "...information concerns selected data - data selected with respect to problem, user, time, place, and function".

Schoderbek (1968: 43) also emphasizes the role of information as reducing uncertainty. Other authors have emphasized other characteristics of information. To Wilson and Wilson (1965: 22), information is the "capacity for increasing knowledge" and to Boutell (1968: 115) it is a "significance derived from the data". McDonough (1963: 71-72) defines information as a label for:

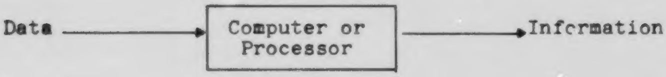
> evaluated data in a specific situation. When the individual singles out one of his problems and finds among his data materials that help him solve the problem he is converting or isolating information from data...(It is a) change from data to information when it is put to use in making a decision.
> The distinction between data and information gives opportunity to create classifications that can be used for further analysis.

McDonough and Garrett (1967: 4) discuss the transformation of information as the transmission of "data to 'relevant' people in the organisation informing them and thereby becoming 'information'. Until it reaches this last stage in a system, data is only potential information".

The definition of information as a transfer of data through processing is illustrated in Figure 4.

Figure 4

TRANSFORMATION OF DATA INTO INFORMATION

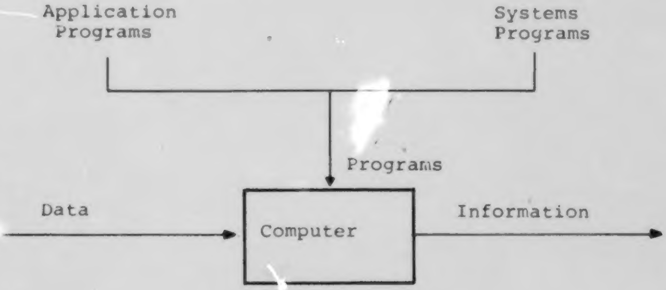


In the example cited earlier, processing involved the relating of a number to a name and the decoding of a sex code into a sex condition. In other cases, the processing may require arithmetic or mathematical operations, data reorganisation, and other processing of data.

The processing is governed by instructions for processing started in a form that is acceptable to a computer. These instructions are called a program, more specifically an applications program since it is concerned with one specific type of application. The instructions however are written in an English-like programming language like COBOL, or a partly mathematical programming language like FORTRAN. These languages are not directly understood by the computer equipment that only recognizes electrical signals. The programming language must then be translated into electrical signals. This is done by yet another set of programs called systems programs. The two types of programs transform the data into output. This relationship is shown in Figure 5.

Figure 5

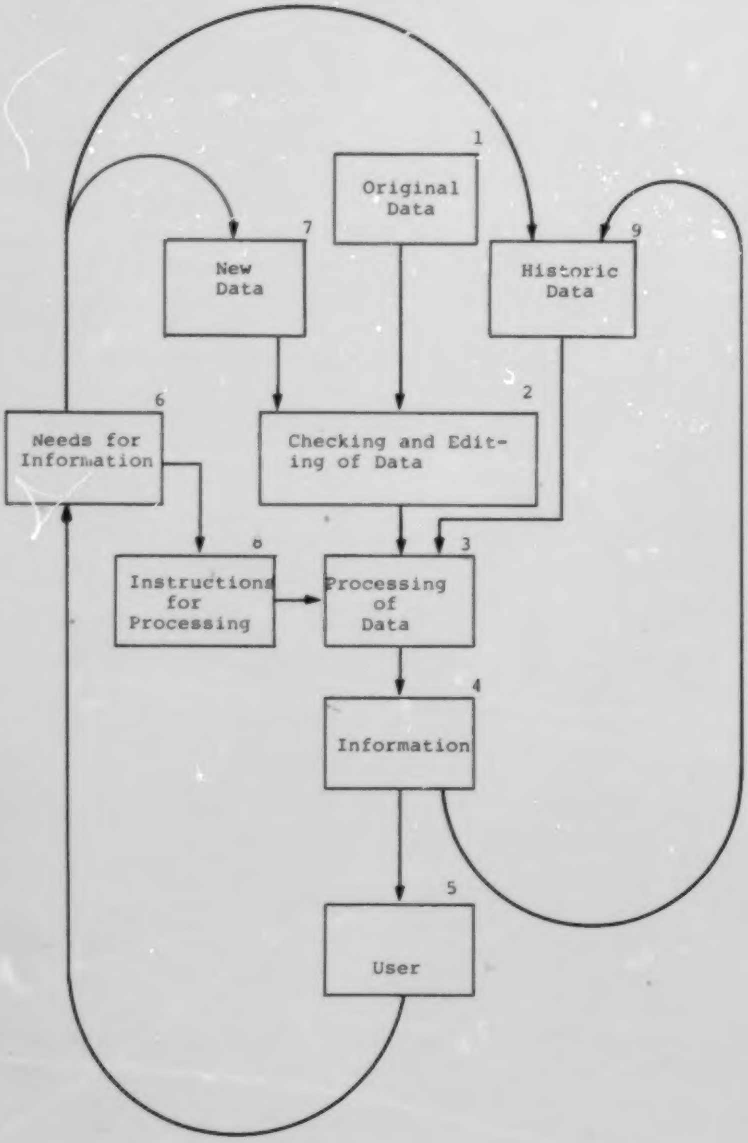
USE OF PROGRAMS FOR TRANSFORMING DATA INTO INFORMATION



The production of information implicitly requires a user, the one who identifies the information needed and uses it. After the information is used, it may again become data which, along with other data, is reprocessed and the cycle of information production continues. This cycle is depicted in the block diagram of Figure 6.

Figure 6

PRODUCTION OF INFORMATION


1
Original Data
7
New Data
9
Historic Data
6
Needs for Information
2
Checking and Editing of Data
8
Instructions for Processing
3
Processing of Data
4
Information
5
User

One could consider the start of the cycle the original data (box 1) that is checked and edited (box 2). If necessary, the data is modified to ensure that it is complete and has no apparent errors and inconsistencies. The data is then processed (box 3) and information is generated (box 4). The information is then sent to the user (box 5). The satisfaction or dissatisfaction concerning this information is then fed back, which may modify the needs for information (box 6). The modified needs for information then determine the new data (box 7), which is used for the next cycle of reprocessing, again with the checking and editing (box 2). The modified needs for information (box 6) also affect the instructions for processing (box 8), which in turn determine future processing.

The information once generated (box 4), in addition to being sent to the user (box 5), also becomes part of the historic data (box 9). An example would be the payroll for the month. Social security and other deductions will be carried over as historic data for the following month. The nature of the historic data needed will be determined by the modified needs for information as shown by the arrow between box 6 and box 9. This data is used for processing (box 3) and need not be edited or checked for validity (box 2) since it has already once gone through that stage.

4. AN INFORMATION SYSTEM

Also shown in Figure 6 are most of the basic components of a system. It has input (box 1, 7, 9), processing (box 3; box 2 could also be considered as processing and will be discussed later), control and feedback (box 5, 6, and 8), and output (box 4). If the system has a "wholeness" because it has a common plan and objective, then Figure 6 represents a system, or more specifically, an information system - a system that produces information.

More specifically, one can define an information system as being an assemblage or combination of things or parts forming a complex or unitary whole to produce information according to a plan.

5. SUBSYSTEMS

Subsets of systems with certain properties are called subsystems. They too have components similar to systems, but their objectives are restricted to functions and subgoals.

Subsystems are part of a hierarchy, or are possibly systems themselves. Thus, in an automobile system an engine is a subsystem, but to an engine mechanic an engine can also be considered a system with its own subsystems such as the cooling subsystem, the electrical subsystem, and the internal combustion engine subsystem. Whether an entity is considered a system or subsystem depends on its location in the hierarchy and on which part of the hierarchy is being examined. For example, in an educational setting, if an educational institution was being studied, it would be a system, but if its state's or country's educational system was being studied, that institution would be a subsystem.

6. INTEGRATED SYSTEMS AND SUBSYSTEMS

With systems divided into subsystems, problems of co-ordination and compatibility between subsystems arise. They should be united and co-ordinated in order to achieve system effectiveness and efficiency. The unification of subsystems in terms of flow of data information within a system is referred to as integration.

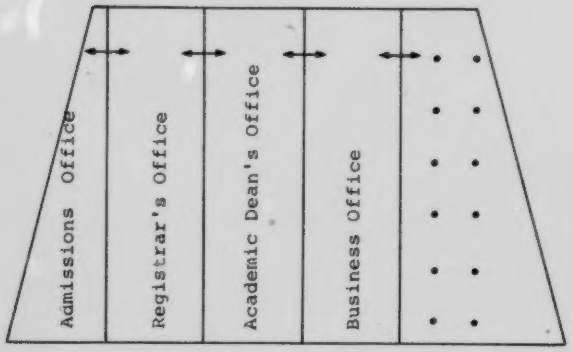
There are many types of integration: horizontal, vertical or functional, and longitudinal. Each of these will be discussed below. A discussion of the techniques of integration will be discussed in another paper on the Data Base.

6.1 Horizontal integration

Integration at one level of administration and management is referred to as horizontal integration. It is illustrated for the operational level of administration in Figure 7 in which only a few organisational units are shown. Historically, each organisational unit, starting with the Business Office and followed by the Registrar's Office had their own information subsystem, including sometimes a computerized system.

Figure 7

EXAMPLE OF HORIZONTAL INTEGRATION



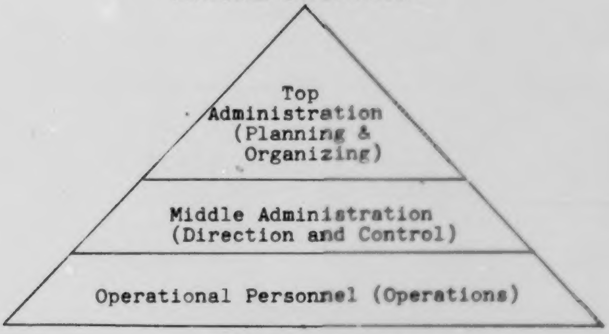
Each of the computerized subsystems had their own computer, their own computer personnel and their own data system. Data and information did not flow or exchange between these subsystems. This resulted not only in unrelated and inconsistent information but also in an inefficient, inaccurate and ineffective system. To overcome these problems, the systems had to be interrelated and integrated with a sharing and exchange of data and information between the subsystems. Such a flow is shown by the horizontal arrows in Figure 7 (only flows between adjacent offices are shown) which gives the name horizontal integration.

6.2 Vertical integration

Thus far, only one level of management (an administrator) was discussed. In all institutions there are at least three levels performing different functions as depicted in Figure 8.

Figure 8

VERTICAL INTEGRATION



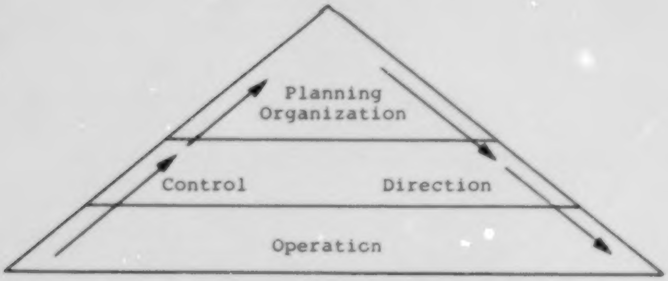
For effective management, data and information must flow between the levels of management to perform the different functions of operation, control, planning, organisation and direction. This flow is depicted in Figure 9. Since the flow is vertical in the organisation the integration resulting from such flow is often referred to as vertical integration(1).

---

1) Unfortunately, the term vertical integration is referred to another phenomena of integration; that of relating all processes of information production, i.e. from data generation through information usage. For a detailed discussion see Hussain (1973: 74).

Figure 9

FLOW OF INFORMATION



6.3 Longitudinal integration

Another dimension of integration is the time dimension. We are interested not only in system performance as of today, but as of yesterday, and even five or ten years ago. This information is needed for record purposes (such as producing the educational transcript of a student) and for predicting future values of variables such as student enrollment, staffing patterns, and curricula programs. Such integration is known as longitudinal integration.

7. TOTAL SYSTEMS AND MIS

A system can be integrated horizontally, vertically and longitudinally. Such a system is called a total system. It is similar in concept to an MIS, a Management Information System which has been defined by Aaron (1969: 213)(1) as being "an information system which provides the manager with the information he needs to make decisions".

The existence of an MIS is questioned by many, including Canning (1969: 1), "The term 'Management Information System' is so ambiguous that one could draw almost any conclusion one wants about such systems. Defined in one way, they do in fact exist today; defined in another way, their realisation is not even close".

As for total systems, some writers in the field (Ackoff, 1967) feel that there are few if any and that they are not feasible because the necessary effort and technology is not available (Brooker, 1965)(2).

---

1) For other definitions, see Colbert (1967: 4), and Dickinson (1968: 8). For a survey of five other definitions, see Canning (1969: 5).

2) For a critical view, see "MIS is a Mirage" (Dearden, 1972). For an optimistic view, see "The Future MIS" (Kriebel, 1972).

Crowley (1968: 273) is more emphatic: "...I do visualise total integrated systems that will collect, digest and deliver decision-inspiring data to management, but only that which can be generated from the flow of activities carried on by the firm's own employees, that is data relating to inventory, availability and price; manpower availability and cost; sales and deliveries, and so on.

I do not visualise total integrated systems that will synthesize for management decision purposes all the facts relating to ambient political, economic, and other factors that the management would like to have at its fingertips in order to remove all hazards from business projections and decisions."

The impracticalities of achieving a total system or an MIS, however, should not preclude the desirability for planning in that direction. Schoderbek (1968: 109) argues for planning for "a total system approach even though such a procedure appears to be a highly ambitious project and to approximate utopian proportions. Unless this approach is pursued there will be much duplication of inputs at diverse hierarchial levels and in functional areas as well". The concepts of integration crucial to a total system or an MIS are valued in themselves and should be attempted in a planned and deliberate manner.

8. SUMMARY AND CONCLUSIONS

In this chapter, a system is defined and its components discussed. The components include the physical components (input, processor, and output); the goals and objectives to be achieved (provided as input); the feedback and control necessary to make performance conform to the objectives; and the human components, which include administrators who set the goals, controllers, sensors, and workers who operate the system. These components are designed as a "whole" to make a system work according to a development plan.

The term, system, connotes order and organisation. It represents a frame of mind and framework of the interrelated whole for solving problems. A system can sometimes be composed of subsystems where a subsystem consists of related activities. There is a hierarchy of systems and subsystems; some subsystems are also systems with their own subsystems. Whether a system is considered a system or a subsystem depends on which part of the hierarchy is being studied.

There are many systems, one being an information system in which information is the output and data is the input. Here data is defined as a set of characters to which a meaning can be assigned; information is selected data that is processed.

A very important concept discussed is integration of systems. Three types of integration are explained: horizontal, vertical, and longitudinal. Integration is the co-ordination of subsystems to prevent overlapping and duplication. In horizontal integration, the system is integrated at one level, such as operations, control or top decision-making. In vertical integration, a system is integrated at all levels of management and decision-making. Longitudinal integration integrates with what has occurred in the past because this is necessary for record-keeping and making predictions for the future.

The total system is defined. A total system is integrated horizontally, vertically, and longitudinally and includes the totality of related variables. A similar concept to the total system is an MIS - a Management Information System. It is one that provides all information required by management. Many practitioners argue that neither total systems nor MIS exist anywhere let alone in higher education because we do not have the necessary technology nor do we devote the effort necessary. However, we can and must work and plan towards such systems. Meanwhile the underlying basic concepts and techniques are valid and important even for partially integrated systems.

The subject of this chapter is both broad and complex. Therefore, to help the reader interested in pursuing the subject further there are three annotated bibliographies at the end of the book. The first is specifically on basic concepts of information systems - the subject of this chapter. The second is on text books on Computers and Information Systems. The third and final annotated bibliography is on references specific to educational management.

Whether the information system is integrated or not, it has a cycle - a set of steps that must be followed in the development of any information system. This life cycle is the topic of the next chapter.

REFERENCES

Aaron, J.D., (1969) "Information Systems in Perspective", *Computing Surveys*, No.r, December.

Ackoff, R.L., (1967) "Management Misinformation Systems", *Management Science: Applications* (Series B), XIV, No.4, December, B-147.

Boutell, W.S., (1968) Computer-Oriented Business Systems, Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Brooker, W.M.A., (1965) "The Total Systems Myth", *Systems and Procedures Journal*, XVI, No.4, July-August, pp.28-32.

Canning, R.G., (1969) "What is the Status of MIS", *EDP Analyser*, VII, No.10, pp.1-12.

Colbert, B.A., (1967) "The Management Information System: Pathway to Profit", *Price Waterhouse Review*, Spring.

Crowley, W.J., (1968) "Can We Integrate Systems without Integrating Management", in W.F. Boore and Jerry R. Murphy, *The Computer Sampler: Management Perspectives on the Computer*, New York: McGraw-Hill Book Company, Inc., pp.269-281.

Dearden, J., (1972) "MIS is a Mirage", *Harvard Business Review*, January-February, pp.90-99.

Dickinson, G.W., (1968) "Management Information-Decision Systems", *Business Horizons*, XI, No.6.

Johnson, R.A., F.E. Kast, and J.E. Rosenzweig, (1967) "The Theory and Management of Systems (2nd Ed.), New York: McGraw-Hill Book Company, Inc.

Kriebel, Charles H. (1972) Business Automation, (Now Infosystems), June, pp.18-19.

McDonough, A.M. (1963) Information Economics and Management Systems, New York: McGraw-Hill Book Company Inc.

McDonough, A.M. and L.J. Garrett, (1967) Management Systems Working Concepts and Practices, Homewood, Ill.: Richard Irwin, Inc.

Schoderbek, P.P., (1968) ed., Management Systems: A Book of Readings, New York: John Wiley & Sons, Inc.

Wilson, I.G. and M.E. Wilson, (1965) Information, Computers and System Design, New York: John Wiley & Sons, Inc.

Chapter 3

LIFE CYCLE OF AN INFORMATION SYSTEM(1)

by

K.M. Hussain

CONTENTS

1. Introduction 30
2. The Development Process 30
    2.1 Feasibility study 30
    2.2 Determining systems requirements 30
    2.3 Design and implementation of system 32
        2.3.1 Data base 32
        2.3.2 Physical Preparation 32
        2.3 3 Procedures 32
        2.3.4 Programs 33
    2.4 Organizational changes 33
    2.5 Testing the solution 33
    2.6 Conversion 34
    2.7 Project management 34
3. The Redevelopment Cycle 34
4. Need for Understanding the Development Process 36
5. Participation by Administrators 37
6. Summary and Conclusions 40

References 42

Annotated Bibliography 316

---

1) This chapter is partly an extract from Development of an Information System for Education by Khateeb M. Hussain (c) 1973. It is reprinted by permission of Prentice-Hall, Inc., Englewood Cliffs, New Jersey, United States.

1. INTRODUCTION

This chapter is largely a summary and overview of the many other papers relating to the process of developing an information system. Each development consists of a set of stages that must be carefully planned and executed.

The stages of development are summarised with special emphasis on their interrelationships. This is followed by a discussion of the need for an understanding of the development by different levels of administrators, and the need for their participation in the development process.

2. THE DEVELOPMENT PROCESS

The "gross" stages of system development are shown in the network diagram in Figure 1. Each stage consists of a set of activities, which are displayed and discussed in detail in separate papers. The stages are summarised below:

2.1 Feasibility study

The first stage in the systems development cycle is the feasibility study (activity 10-20 in Figure 1). This is the examination of possible alternate solutions to the systems development problem and the checking of these solutions for constraints of technology, resource availability, and organizational structure. For each solution that satisfies the constraints (and is feasible) the costs are compared to the benefits. If the costs are greater than the benefits or if the organization cannot afford the resources required, then the development process is terminated. If the benefits are equal to or greater than the costs, then one of these solutions is chosen for implementation.

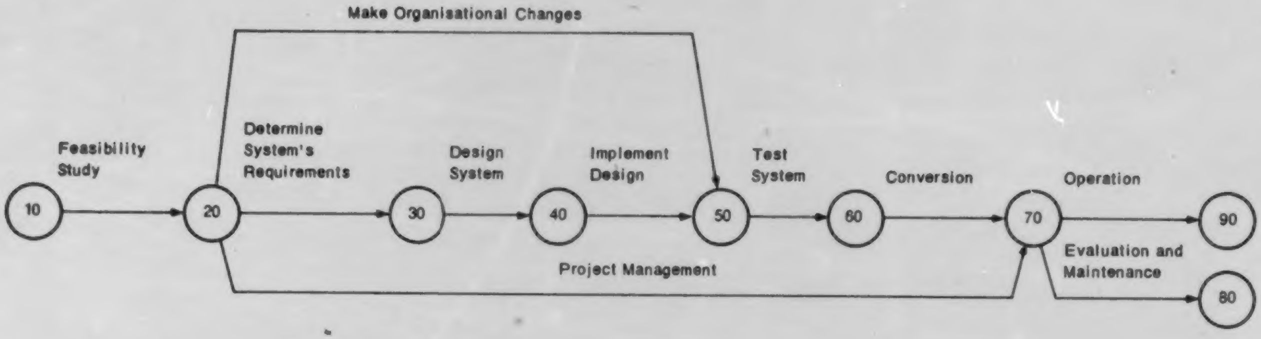
2.2 Determining systems requirements

Following the decision to implement a solution, the problem needs to be defined more precisely than it was in the feasibility study. This is shown as activity 20-30 in Figure 1.

In this stage, the objectives, policies, and constraints of the user must be stated completely and in operational terms. These statements are the responsibility of the user (the administrator or manager).

Figure 1

DEVELOPMENT AND REDEVELOPMENT OF AN INFORMATION SYSTEM


Make Organisational Changes
Feasibility Study
Determine System's Requirements
Design System
Implement Design
Test System
Conversion
Operation
Evaluation and Maintenance
Project Management
10
20
30
40
50
60
70
90
80

2.3 Design and implementation of system

The systems requirement statement is the basis on which a systems designer creates and designs the system. He produces a "blueprint" which will be the basis for the production of the new system.

The design is a set of detailed specifications for each of:

1. Data Base
2. Physical Preparation
3. Procedures
4. Program Solutions

Each of these design specifications is produced and implemented in parallel and then assembled into a complete system. The design and implementation are shown as activities 30-40 and 40-50 respectively in Figure 1. They are discussed (in terms of the detailed specifications) briefly below:

2.3.1 Data base

The data base is crucial to any information system. Too much data would lead to inefficiency in storage and processing; too little data would lead to ineffectiveness because the desired output would not be available. The contents of the data base must be determined after a careful analysis of the output needs and must be responsive to changing environment and changing needs of the user. The organization and storage of the data must then be designed so that it is accessible and retrievable.

2.3.2 Physical preparation

The physical preparation for a system is mainly the selection and installation of equipment (computer equipment or even input equipment). This usually involves technical decisions made by specialists and consultants, but these decisions often concern the administrator, inasmuch as equipment requires a large commitment of funds (one-time and recurring costs), as well as an allocation of space.

2.3.3 Procedures

A procedure provides information on how, why, and when a job is to be performed. Most systems have procedures which must be designed and tested carefully because they often determine both the effectiveness and the efficiency of the system. The prime responsibility for the implementation of procedures is that of the user, who must eventually use the procedures.

2.3.4 Programs

One of the design specifications is concerned with the output and solutions that the new system must generate. Sometimes manual methods are used to arrive at solutions. In many cases, however, the problem may be complex enough to require the use of a computer. In this case the computer must be instructed as to how to produce the desired solution. These instructions are called a program.

The computer programs are written by professional programmers. It is important, however, for the administrator to know programming capabilities and limitations, and their relationship to the overall systems development process.

2.4 Organizational changes

New systems often require that jobs be performed differently and by different personnel. This may require structural changes in the organization, and may result in displacement and unemployment. The adjustments to the systems changes must, therefore, be carefully planned and patiently implemented. The failure to do so will result in organizational resistance and the failure of the new system.

A related set of activities concerns orientation and training. Personnel using the new system must be trained and the administrators involved must be oriented as to the "how" and "why" of the new system. This orientation and training requires selection of personnel, preparation of educational media, and the conducting of the orientation and training sessions.

Organizational changes have a long period of gestation. Hence, the changes must begin soon after the feasibility study but before the system is ready for testing. This is shown as activity 20-50 in Figure 1.

2.5 Testing the solution

Once the solution is programmed, procedures prepared, physical preparation completed, and organizational changes implemented, the system is ready for testing. The testing must be carefully planned before it can begin.

There are many approaches to testing. The new system could be tested in parallel to the old, or it could be tested by a pilot system on a small scale. Testing could also be done in different stages and at different levels. Statistical techniques enable the testing of sample data, from which inferences can be made about the whole system. This inference, or the results of the testing of the population, is then compared with the specifications stated in the Determination of the Systems Requirement stage. If unsatisfactory, the system is re-developed. If satisfactory, the system is then ready for operation.

2.6 Conversion

The conversion phase is the converting or phasing-out of the old system and phasing-in to the new system. It involves the converting and changing of all the old procedures, manuals, forms, files, equipment, and sometimes current personnel in favour of ones required by the new system. This phase is what Kirk (1973: 166) calls "probably the most traumatic". It is often the start of new relationships and many an adjustment.

2.7 Project management

A systems development with a large number of interrelated activities is organized as a project. Its activities must be planned, scheduled, and controlled if it is to be completed within the desired time and within the resources allocated for the project. To accomplish this, project management techniques such as the GANNT chart, the Critical Path Method (CPM), or the Program Evaluation and Review Technique (PERT) are used.

Project Management is often initiated if a solution is selected in the feasibility study, and continues until the end of the conversion phase. This is shown by the activity 20-70 in Figure 1.

3. THE REDEVELOPMENT CYCLE

The development process examined thus far has been concerned with all the stages of development, from the initiation of a system study to the time the system becomes operational. Once the system is operational (activity 70-90) it should be evaluated (activity 70-80) (1) in Figure 1. If the evaluation proves that the system is unsatisfactory and the resources for maintaining the system are available, then the system is redeveloped. Once redeveloped, the system is re-evaluated; the cycle of evaluation and redevelopment continues as long as the system is in existence.

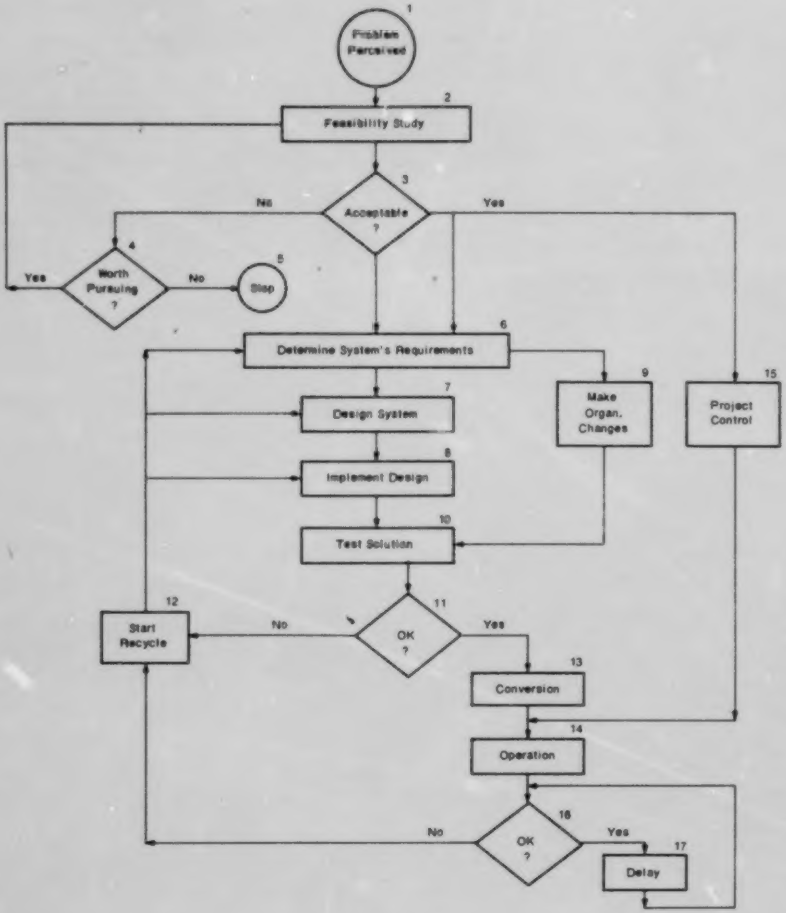
The maintenance and redevelopment process starts with a logical decision: is the system satisfactory? Logical decisions cannot be identified by a block diagram or a network diagram such as Figure 1, but can be shown in a flowchart. Also, some of the stages of initial development shown in Figure 1 involve logical decisions. (Is the proposed system feasible? Are the test results satisfactory?) All these decisions can best be depicted in a flowchart and are shown in Figure 2.

---

1) The activity 80-90 is a dummy and uses no resources including time. Its purpose is to allow unique numbering identification of parallel activities such as 70-80 and 70-90.

Figure 2

FLOW-CHART FOR SYSTEM DEVELOPMENT AND REDEVELOPMENT PROCESS


1
Problem Perceived
2
Feasibility Study
3
Acceptable ?
No
Yes
4
Worth Pursuing ?
Yes
No
5
Stop
6
Determine System's Requirements
7
Design System
8
Implement Design
9
Make Organ. Changes
15
Project Control
10
Test Solution
11
OK ?
No
Yes
12
Start Recycle
13
Conversion
14
Operation
16
OK ?
No
Yes
17
Delay

The perception of the need of a new system (circle 1 in Figure 2) may come from the administrator or the systems analyst. The need initiates a feasibility study (box 2 in Figure 2). The proposed system is tested for feasibility (box 3). If unsatisfactory (NO exit from box 3), it is examined for potential of further study (box 4). If the proposed system is totally unfeasible, then the effort is terminated (circle 5). An example of this case would be where the minimum resources required for a minimum system are more than the resources available. If, however, there is a possibility that another systems design, perhaps with revised objectives and constraints, would be feasible, then another feasibility study is conducted (box 2) and the process recycles (from box 4 to box 2 and then to box 3 and back to box 4) until the proposed study is either considered to be feasible (YES exit from box 3) or terminated (at circle 5).

If the feasibility study is approved (YES exit from box 3), then the stages of determining system requirements (box 6) through the testing of the solution (box 10) and the parallel set of activities on organizational changes (box 9) are executed. These correspond to the activities shown in the network diagram in Figure 1 and will not be discussed further. If the test (box 11) proves unsatisfactory (NO exit from box 11 in Figure 2), then the process may begin to recycle (box 12). The recycle will start with a minor change in the implementation procedures (box 8) or the design of the system (box 7) or even a redetermining of the systems requirements (box 6). The system is then tested (box 11) and recycled till the test is satisfactorily performed (YES exit from box 11). Then after conversion from the old system to the new system (box 13), the system is operational (box 14). Till the system is operational, if it is a large or complex system, it must be managed under project control (box 15).

Once the system is operational, it has to be evaluated and possibly redeveloped. The evaluation (box 16) could be satisfactory (YES exit from box 16) in which case there is a prescribed time delay (box 17) and the system is re-evaluated (box 16). If, however, the evaluation is unsatisfactory (NO exit from box 16), then the redevelopment starts (box 12). As with the development recycle, it may start at any of the boxes 6, 7 or 8 depending on the nature of the problem involved. The recycling may even start at the feasibility study (box 2) if the situation changes radically and requires another feasible study.

4. NEED FOR UNDERSTANDING THE DEVELOPMENT PROCESS

There are many administrators in education (and managers in business) who are disappointed in their information systems and complain about the large gulf between what was promised and what has been achieved. Two reasons contribute to this credibility gap: first, systems analysts often promise more than they can produce; second, the administrator, in his ignorance of information systems, often expects too much. He thinks the information system, especially a computer system, is a magic machine that will produce instantly whatever is requested. This misconception on the part of the administrator and the over-enthusiasm of the systems analyst reinforce each other, resulting in a feeling by the administrators of being let down. The administrator can reduce, if not eliminate, this possibility by participating in the development of his system. A knowledge of system development and his participation in it will not only ensure that he is not oversold but will also reduce the danger of his levels of expectation being exaggerated. Furthermore, an administrator's knowledge and participation in the development of his system is the only way for him to ensure that he gets the information he really needs.

5. PARTICIPATION BY ADMINISTRATORS

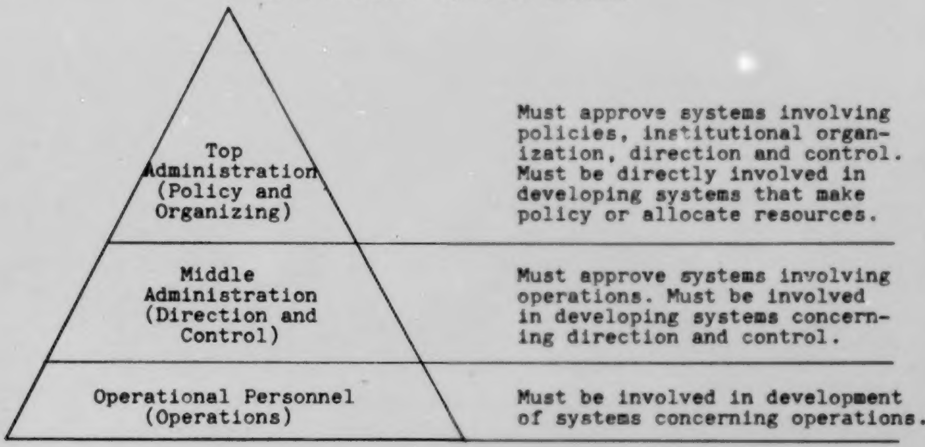
A theme that will be repeated throughout many a discussion on system development is that administrators (or managers) must approve and be involved in the various stages of development. The question that arises is, what level or levels of administration should approve and be involved in which functional level of system development? Generally speaking, the answer is that personnel should be involved in the development of a system when it concerns their functional level, and administrators must approve the development of systems one level below them, except in the case of top administrators, who must also approve systems at their own functional level. This general set of rules is shown diagrammatically in Figure 3.

In addition, the development must be approved by the appropriate institutional committees responsible for co-ordinating such developments (to be discussed later).

Figure 3 implies that systems which only affect operational matters do not need the approval of top-level administrators. An example would be the case of a housing office that spends $20,000 per year in assigning rooms to students. If its functions can be automated within the $20,000 per year annual allocation, then top administration need not be involved, since the system does not involve institutional policies or additional institutional resources. The approval of the middle-level administrator (such as the Director of Housing) in this case would be sufficient. However, if the system requires additional resources (even on a one-shot basis), affects institutional policies or the organization, violates institutional constraints, or creates interdepartmental rivalries, then the approval of top administrators becomes necessary.

Figure 3

LEVELS OF ADMINISTRATION AND OTHER PERSONNEL FOR DIFFERENT TYPES OF SYSTEMS


Top Administration (Policy and Organizing)
Middle Administration (Direction and Control)
Operational Personnel (Operations)

Must approve systems involving policies, institutional organization, direction and control. Must be directly involved in developing systems that make policy or allocate resources.

Must approve systems involving operations. Must be involved in developing systems concerning direction and control.

Must be involved in development of systems concerning operations.

The importance of participation in systems development by the administrator in education corresponds to the importance of participation by managers in business and industry. This necessity of manager participation has been borne out by empirical evidence from a survey done by Garrity, who studied 27 companies in 13 different industries that used computerized information systems. Garrity identified two groups of companies: one group that was unmistakably successful in its use of computer systems, and another that had marginal success at best. In analysing the characteristics of success of the successful "lead" companies, Garrity (1963: 10) observes that:

> Executive management devotes time to the computer systems program in proportion to its cost and potential and in relation to the executives' other responsibilities. This time is spent not on the technical problems, but on the management problems involved in integrating computer systems with the critical management process of the business. In particular, top management time is spent in reviewing the plans and programs for the computer systems effort and then in following up on the results achieved. Through formal progress reviews (taking place at least quarterly), lead company executives ensure that the computer effort focuses on high-leverage applications. They also see to it that current projects are proceeding on schedule. On projects that are not panning out, they look for the trouble spots and institute cor-

> rective action. In short, the lead company corporate executives are personally involved in making the computer pay off ... in every lead company and in two-thirds of the average companies, the computer executive is no more than two levels below the chief executive. (In one-third of the average companies, however, this executive is three levels down). From this it would appear that so long as the computer executive is positioned high enough to have corporate stature (i.e. within two levels of the chief executive), the computer systems effort can succeed - provided, of course, he plays an effective role and is given the kind of top management leadership already discussed.

Even at the operational level, top management has an important influence on the degree of participation. Garrity (1963: 12) states:

> Obviously, the close involvement of operating management is essential if a company is to apply computer systems effectively to inventory management, equipment scheduling, demand forecasting, and the like. The lead companies have achieved it by a combination of factors, including productive missionary work by the technical staff. But top management's attitude seems to be the principal ingredient. Where top management has fostered a tradition of effective line-staff relations, where top management has created an atmosphere favourable to an innovating, inquiring approach, operating executives have been much more willing to participate in the effort. Indeed, in some cases they have been the prime movers.
>
> But over and above this indirect encouragement, top management in most lead companies has specifically spelled out to operating executives the corporate commitment to the computer effort, its objectives, and operating management's responsibility for achieving the anticipated return. One operating manager commented: "I did not volunteer to be a guinea pig for the computer. But the message came through loud and clear. I was expected to use the computer and show results."

The decisions made by administration on matters of information system development would be made by either individuals or groups of administrators. Group decision-making is often preferred because it has the advantage of better representation of the many departments in the organization that are most affected by system changes. This representation will not only improve the probability of organizational acceptance but will ensure a more equitable distribution of systems effort in the organization.

Group decision-making on system development in education should include teaching staff. This not only ensures teaching staff representation (they are important users of an information system) but often draws upon a source of competent and knowledgeable personnel.

The composition of the group deciding on information systems development could vary, depending on the functional nature of the system and the importance of the system. In some institutions, it is a standing committee that maintains the continuity necessary for developing integrated systems, especially of the operate-now-integrate-later type systems. The name of such a committee varies from institution to institution. One is a "Systems Development Policy Committee". The emphasis here is on policy-making, overall co-ordination, and surveillance, rather than the detailed user specification and detailed implementation which is done by an ad hoc design committee or a development committee.

6. SUMMARY AND CONCLUSIONS

The life cycle of an information system consists of a number of stages. Some of these are the responsibility of the analyst. In the other stages the manager must actively participate with the analyst and make the many decisions necessary in defining and implementing the system. He must not abdicate this responsibility because it could result in potentially significant adverse consequences.

The stages of development and the respective prime responsibilities of the analyst and manager are identified below.

Stage of Development	Person Responsible
Feasibility Study	Analyst and Manager
System Requirements	Manager and Analyst
Design System	Analyst
Implement System	Analyst
Test System	Analyst and Manager
Organizational Changes	Manager
Conversion	Analyst
Evaluation and Maintenance	Analyst and Manager

The stages of development will vary somewhat with institutions and designers. The sequence of some stages and their names and terminology may vary. One author (Murdock, 1970: 23-24) identifies 17 different schemas. Yet another one is expressed in poetry by a secretary in an American firm (Driscoll, 1968: 318).

(Canto the first: Proposal)

"An information system", said the president, J.B.,
"Is what this company sorely needs, or so it seems to me:
An automated, integrated system that embraces
All the proper people, in all the proper places,
So that the proper people, by communications linked,
Can manage by exception, instead of by instinct."

(Canto the second: Feasibility study)

They called in the consultants then, to see what they could see,
And to tell them how to optimise their use of EDP.
The consultants studied hard and long (their fee for this was sizable)
And concluded that an information system was quite feasible.
"Such a system", they reported, "will not only give you speed,
It will give you whole new kinds of information that you need."

(Canto the third: Installation)

So an information system was developed and installed
And all the proper people were properly enthralled.
They thought of all the many kinds of facts it could transmit
And predicted higher profits would indeed result from it;
They agreed the information that it would communicate
Would never be too little, and would never be too late.

(Canto the last: Output)

Yet when the system went on line, there was no great hurrah,
For it soon became apparent that it had one fatal flaw;
Though the system functioned perfectly, it couldn't quite atone
For the information it revealed - which was better left unknown.

REFERENCES

Driscoll, Marilyn, (1968) "An Information System", The Arthur Young Journal, (Winter).

Garrity, John T., (1963) "Top Management and Computer Profits", Harvard Business Review, Vol.XLI, No.4.

Kirk, F.G., (1973) Total System Development for Information Systems, (New York: John Wiley and Sons).

Murdock, R.G., (1970) "MIS Development Procedures", Journal of Systems Management, (December), pp.22-26.

Part Two

DEVELOPMENT OF AN INFORMATION SYSTEM

This Part is concerned with the development of an information system. But only a few selected stages amongst those identified in the last chapter will be discussed. These are: the development of a data base which is part of the design stage (Chapter 4); the approaches to design of the data base as well as the other components of the system (Chapter 5); and finally, the testing and conversion of the system (Chapter 6). The remaining stages of development have been deleted because they are well-documented elsewhere - both in the literature on business information systems, the principles of which are applicable to the higher education environment, and in the literature on information systems in higher education.

Chapter 4

DATA BASE: DESIGN AND IMPLEMENTATION

by

K.M. Hussain

7

CONTENTS

DESIGN

1. Introduction 47
   1.1 The Audience 47
   1.2 Scope 48

2. Some Basic Concepts 48
   2.1 Data and Data Elements 48
   2.2 Hierarchy of Data 51
   2.3 Classification of Data by Origin 53

3. Data Elements 55
   3.1 Transaction Data Elements 55
   3.2 Supporting Data Elements 55
   3.3 Reference Data Elements 55
   3.4 Identification Data Elements 56
   3.5 Linking Data Elements 56
   3.6 Checking Data Elements 56
   3.7 Security 58
   3.8 Data Elements for Control 58
   3.9 Data Elements for Planning 58
   3.10 Other Data Elements 58

4. Data Element Dictionary 59
   4.1 Why a Data Element Dictionary 59
   4.2 Designing a DED 59
   4.3 DEDs and a Common Data Base 62
   4.4 Uses of a Common Data Base 65

5. Specification of a DED 66
    5.1 Selecting Data Elements 66
    5.2 Description of Data Elements 69
        5.2.1 Name 69
        5.2.2 Definition 69
        5.2.3 Derivation Algorithm 70
        5.2.4 Units 71
        5.2.5 Format 72
        5.2.6 Width of Field 73
        5.2.7 Validity Rules 73
        5.2.8 Identification 74
        5.2.9 Status 75
        5.2.10 Other Descriptors 75
    5.3 Contents of the DED 75

6. The Program Classification Code 77
    6.1 Introduction 77
    6.2 The PCS 77
        6.2.1 Construction of the PCS 77
        6.2.2 Structure of the PCS 78
        6.2.3 Code Format of the PCS 78
    6.3 Use of the PCC 78

IMPLEMENTATION

7. Uses of a Dictionary/Directory System 83
    7.1 Introduction 83
    7.2 Uses of Management and Other Personnel of the DED/DD System 84
        7.2.1 Use as a Repository of Information 84
        7.2.2 Identifying Uses and Users of Data Elements . 87
        7.2.3 Identifying Data Creators 87
        7.2.4 Identifying Transactions 88
        7.2.5 Descriptors Needed for 7.2.1-7.2.4 88
    7.3 Uses by Computer Software of the DED/DD System . . . 90
        7.3.1 Access Directory 90
        7.3.2 Use in Processing 91
        7.3.3 Locational Information 92
        7.3.4 Use to Generate Test Data 92
        7.3.5 Code File 95
        7.3.6 Descriptors Needed for 7.3.1-7.3.5 95

8. Summary and Conclusions 96

References 100

Annotated Bibliography 328

DESIGN

1. INTRODUCTION

A data base is required by most institutions that operate and plan for future operations. Not only must a data base be created but it must also be maintained. In doing so there are two groups of persons involved. One group consists of the administrators and managers who create the data that goes into the data base and use the results that come out. The second group consists of the data processors. They store the data and process it as needed by the user. The interest of the two groups as they relate to the data base is quite different, and also the people involved have different backgrounds and knowledge. Therefore, any paper on the subject should be addressed to one or the other of the two groups. This chapter is specifically addressed to the administrator group.

1.1 The Audience

The target audience of this chapter, the administrator (or manager) is important in the creation of a data base. He must make the decisions necessary in establishing the data base. The maintenence of the data base, though of some interest to the administrator, is a more technical subject. But in both creation and maintenance the administrator must be involved since the data base provides him with information that is the basis of decision making, and the quality of the information he derives from the data base is very directly a product of the quality of the data base itself. There is a cliché in data processing: GIGO, garbage in, garbage out. Hence, what goes into the base must be prepared with great care, and this is the responsibility of the administrator. He alone has the judgement, knowledge and experience necessary in making many of the decisions required in creating the data base. A technician or analyst, often given this responsibility by delegation or default, is unable to perform this function adequately because he lacks the background and knowledge of the administrator and he does not have the motivation of an ultimate user of the product.

In order for the administrator to perform his function in relation to a data base, he must have a basic understanding of data bases and their components. Providing such an understanding is the objective of this paper. It does not presume any prerequisite knowledge of mathematics or of data processing and computers. It only assumes an open mind and a will to apply oneself to the subject.

1.2 Scope

Since this chapter is not addressed primarily to the technical person, it will not be concerned with the technical aspects of a data base. Instead a part on Design is concerned with the broader conceptual problem and basic concepts of data (sections 2,3); the contents of a data base and its organisation (sections 4,5,6) and a second part, on Implementation, is concerned with the by-products of a data base organisation (section 7). Finally, section 8 is devoted to a summary and conclusions.

2. SOME BASIC CONCEPTS

Most institutions of education require a data base which is an integrated set of data where data is facts or observations. The data when processed with other related data provides information which is the basis of current operations and information on which decisions concerning future operations are based.

Institutions can have more than one data base. As an institution gets larger and more complex its functions and the decision making process require that more data be collected and correlated. Operating this data in a common data base results in efficiency and consistency of processing. It is such a data base - a common data base - that we are concerned with in this paper.

2.1 Data and Data Elements

We mentioned data as being a fact or an observation. Consider the observation and recording of SABLE. By itself SABLE is meaningless unless it is identified as the name of an entity such as a student (or a product). The entity may have attributes with a single value such as sex, marital status, major, or age, or it may have attributes with multiple values such as courses taken, degrees held, grades, etc. An attribute for which data is collected is a data element. Thus the name of a student is the data element and the value of that data element is the data SABLE. Similarly, if a student takes 12 contact-hours, the data 12 is the value of the data element contact-hours. If the 12 units of contact and the student SABLE were related, they would form a logical grouping of data elements yielding the information: SABLE has taken 12 contact-hours. If we associated these data elements with the data elements of when and where the contact-hours were taken we yield more information and thus develop a logical record on the student SABLE.

Some records that are long take more than one page of data, or more than one card used in data processing. These are referred to as physical records. More than one physical record may be one logical record, or conversely, more than one logical record may be recorded on one physical record.

Sometimes a physical record may contain the same data element but with different values. For example, a listing of codes of courses taken by a student or many grades given to a student in a semester. Such data is referred to as a data array, or a list of data in one dimension. If the data is in a matrix of two dimensions it is called a data matrix. The classic example is the Induced Course Load Matrix where each element is the number of students (or contact) for each major taking courses in a discipline or department.

A record consists of data elements. Each data element must be assigned a space in a physical record. This space is referred to as a field. In the case of data processing the physical record is a card and the physical space is measured in columns (technically 80 columns to a card). The columns occupied by a data element are referred to as to the width of the field and must be specified so as to plan for the space needed for records.

Another situation arises when one data element is composed of more than one value, i.e. one data element is a specific permutation arrangement and grouping of other data elements. An example of such a data element is date of birth. Its value may be 010324 with each two digits representing day, month and year in that order. Day, month and year are each data elements of two digits combined to give the data element date of birth. Thus the data element date of birth is a group data element composed of three data elements. In data processing terminology, the date of birth is a field of 6 digits (or six columns on a card) consisting of 3 subfields (day, month and year); each subfield being 2 digits and having a width of 2 card columns each.

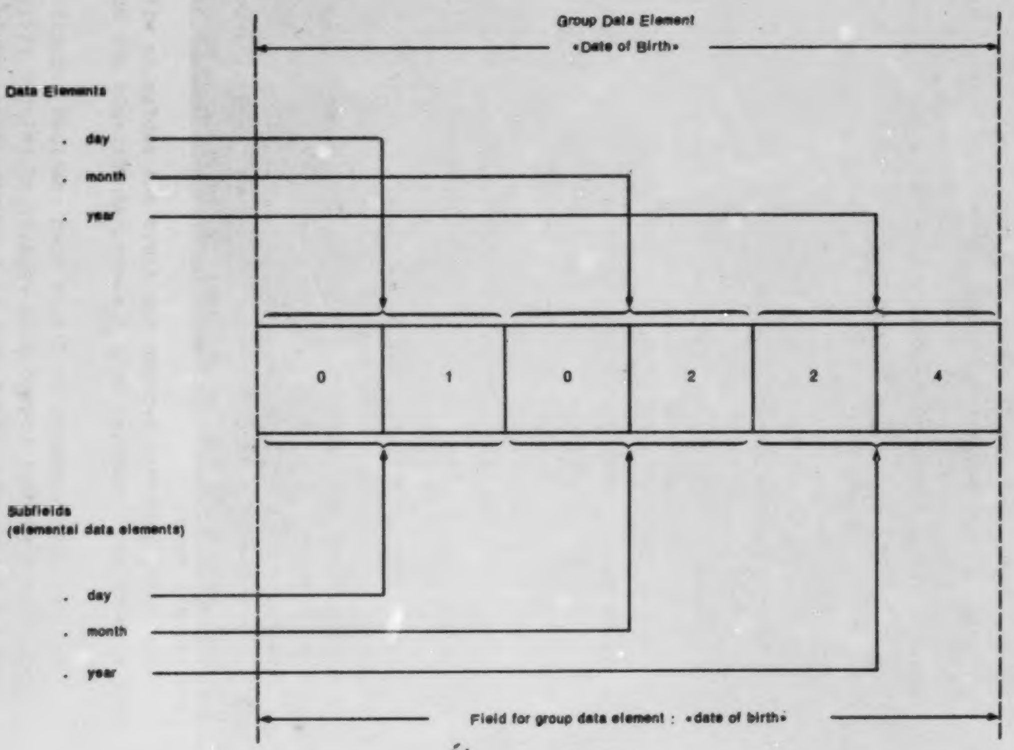
The correspondence between the field and subfields with a group data element and elemental data element respectively are shown in Figure 1.

Related data elements or fields when combined constitute a record. Thus a student record will consist of related fields such as name, address, sex, major, etc.; a course record would include the course title, its coded title, the instructor, the time and place when it is taught, etc.; a personnel record would have the name of each employee, his address, title, salary, etc.; a space record would identify each room in each building, its size, equipment, etc.; a financial record would include a budget and cost categories for each cost centre as well as its accounts receivable, accounts payable, etc.; and so on. The records that are related

48

Figure 1

CORRESPONDENCE BETWEEN FIELDS AND DATA ELEMENTS. ELEMENTAL AND GROUP DATA ELEMENT


Group Data Element
«Date of Birth»
Data Elements
. day
. month
. year
0
1
0
2
2
4
Subfields
(elemental data elements)
. day
. month
. year
Field for group data element : «date of birth»

can then be combined to constitute a file. Thus the records of students will be a student file. Educational institutions typically have five files: a student file, a course file, a personnel file, a space file, and a financial file. Other examples of files are an alumni file, a property file, an auxiliary services file, a police department file, etc. These data files when put together constitute a common data base, but with one important provision; the data files must be integrated, i.e. related or linked together. The linking is necessary to provide a base or foundation that can be shared by all users. Without the linking capability we would have a set of files but not a common data base.

The need and role of linking can best be explained by an example. Suppose that I wanted to calculate the salary cost of a course. I would (or rather the computer program would) use the course file and identify the instructor who teaches the course. Then I would check the personnel file and find out the salary of that instructor. From this file I would also learn the instructor's normal teaching load. Now the salary cost of the course can be calculated. But to calculate the cost these two files must be related or linked; that is, the identification of the instructor in the course file must be exactly the same as the identification of the instructor in the personnel file. This identification may be the instructor's surname with initials or a more unique identification such as his social security number or personal identification number. Without the same identification the computer cannot correlate data from two separate files. One could, of course, put salary information in the course file, or course information in the personnel file, but this would result in much redundancy of data elements when they are not used and a waste of processing resources. The more logical solution is to collect and store related data in one file and then connect files by a common data element. Such a data element is a linking data element. It is crucial in a common data base that is composed of many files.

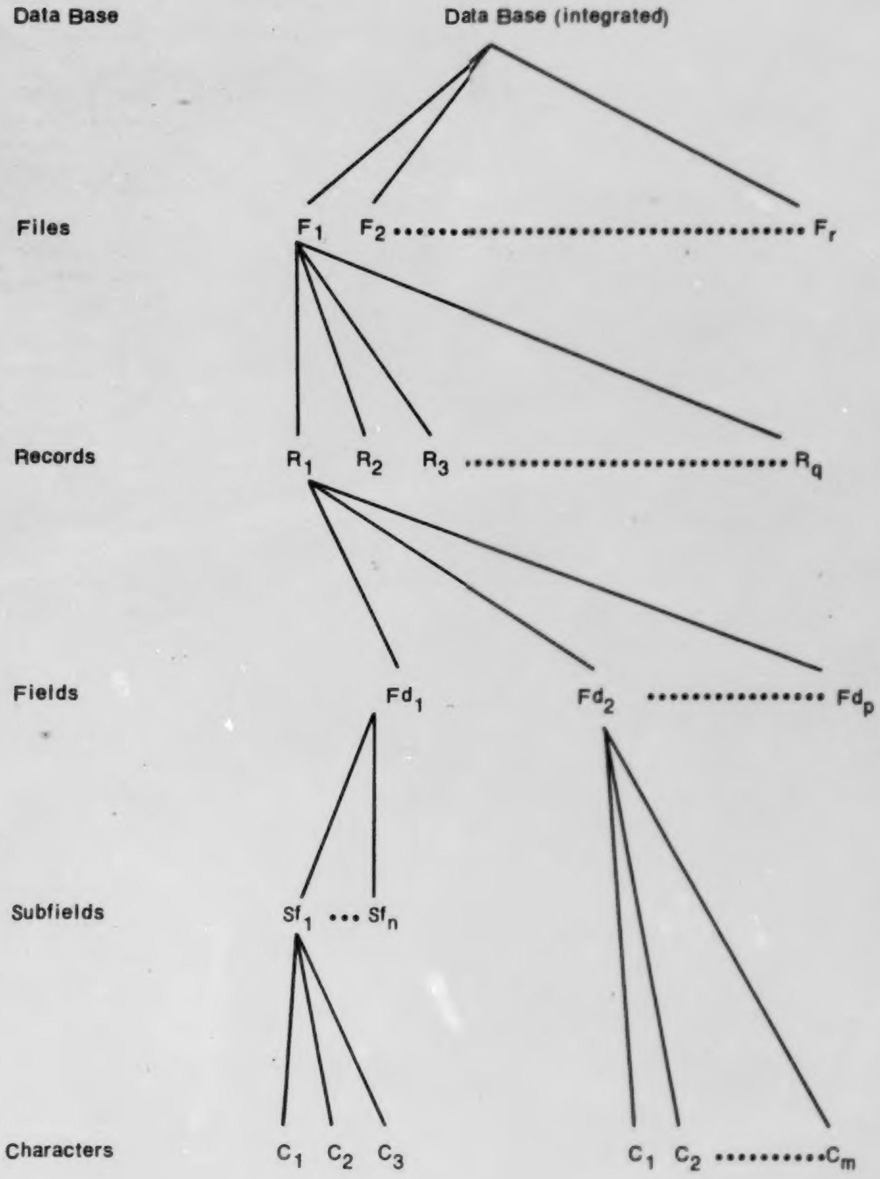
2.2 Hierarchy of Data

We have discussed the hierarchy of the data base: data element - record - file - data base. This relationship is shown diagrammatically in Figure 2.

In Figure 2 we clearly identify that some fields have subfields and some do not. Each field or subfield would correspond to a data element. The number of subfields constituting a field, the number of fields constituting a record, the number of records constituting a file, and the number of files in a data base are all undefined. There is theoretically no upper limit on these values (n, p, q, r and s in Figure 2). In practice, however, all these values need to be known to determine how the data will be stored.

Figure 2

HIERARCHY OF DATA


Data Base
Data Base (integrated)
Files
F_1 F_2 .... F_r
Records
R_1 R_2 R_3 .... R_q
Fields
Fd_1 Fd_2 .... Fd_p
Subfields
Sf_1 ... Sf_n
Characters
C_1 C_2 C_3
C_1 C_2 ....C_m

In Figure 2 we also have another level of data, one below the field or subfield element. This level consists of characters which represent the value of the data element. The characters may be numeric, alphabetic, or combined (referred to as alphameric or alpha numeric). They could also be special characters such as +, -, x. In all cases the number of characters that constitute the field or subfield determine the width of the field and must be specified with care. An unnecessarily wide field will result in a waste of computer storage space and processing time. An insufficient width of field will result in the incapability of processing data without restructuring the data base which is monetarily costly and operationally disruptive.

The problem of an insufficient width can be illustrated by examples. Consider the subfield of year discussed in Figure 1. A width of two digits is sufficient for the date of birth assuming everyone is born in this century. But suppose we had a data element for the year of construction of each building on campus for purposes of calculating depreciation. Would two digits be enough? Not for a campus that is over 100 years old. It would require a field of four characters to differentiate between a building constructed in the 1800s and one constructed in the 1900s.

Similarly consider the field width for a data element of the number of students on campus. For a campus of 1-9,999 students a four character field will suffice. But if the campus grows to over 10,000 students then a four character field is insufficient. Adding the extra character would require changing the data record and modifying the relevant computer programs which could be costly. Thus the expected width must be specified with care after allowing for growth.

The characters in a field are important but also crucial in the data hierarchy is the data element. That is what determines the contents of the data base and the contents, in turn, determine the information that comes out of the data base. The definitions of each data element affect the accuracy, adequacy, relevance and value of the output.

We shall discuss the definitions of the data elements presently. Before that, however, we need to discuss one more set of concepts; the origin of data.

2.3 Classification of Data by Origin

Earlier we defined data as being an observation or a fact. SABLE is a fact. It is the value of the data element: name of student. Similarly 12 is data because it is the number of hours of contact taken by the student SABLE and is the value for the data element of number of hours of contact taken. Sometimes data is assigned data, such as a code being assigned to identify a course. All such data is primitive in that it has not been derived from other data elements nor is it a result of any processing. Such data is referred to as raw data.

Raw data is distinct from data that is derived from other data elements. As an example, consider the average class size of the sections of Latin 101. The average cannot be observed nor is it a primitive fact. It is not raw data. But it can be derived from other data elements; i.e. the data element of the number of students in all sections of Latin 101 divided by the data element of the number of classes offered for Latin 101. This then is derived data.

There is another type of data that is neither primitive nor derived. It is an estimated data element. Consider the data element of student enrollment for Latin 102 next semester. This cannot be observed today and is not a fact because it is in the future. This data element must be calculated by a formula such as:

$Y = bX$

where Y = future student enrollment in Latin 102
X = current enrollment in the neighbouring high school
b = coefficient

The coefficient b is the relationship between Y and X, (Y/X), and must be estimated by statistical techniques based on historical data of Y and X, by hunch, or by the administrator's perception of the relationship of Y and X. This data element (b) is an estimated data element.

In this discussion we have extended our original definition of data. Data can now be defined as raw data (an observation, fact or assignment), derived data, or estimated data. This classification is summarised in Figure 3.

Figure 3

CLASSIFICATION OF DATA BY SOURCE

Data
- raw data
  - observation
  - fact
  - assignment
- derived
- estimated
  - hunch
  - guesswork
  - statistical techniques

3. DATA ELEMENTS

Having examined the data hierarchy and having classified data elements by origin, we are now ready to consider the question: what types of data elements do we need? The answer to this question will be a classification of data elements by use, giving reasons and examples in each case.

3.1 Transaction Data Elements

Most data elements in any common data base are created by transaction. For example, the admission and registration of a student are both transactions, which generate a whole series of values for data elements such as name, address, sex, courses to be taken, etc. More transactional data is generated when courses are added or dropped, when grades are assigned, when disciplinary actions are taken (if any), etc. Similarly transactional data is created when property and supplies are ordered, received and paid for, or when staff is hired, work hours are recorded and salary checks are issued. All these data are values for transactional data elements.

3.2 Supporting Data Elements

The values of transactional data may vary over time. In contrast there is related support data that is constant, i.e. data that does not change with every transaction. For example, the letter grades and points assigned to each letter grade (A=4, B=3, C=2, etc.), data used to calculate grade point averages. The grade point set of values is an example of an array data element used to support transactional data elements. Other examples of supporting data elements are teaching loading weights for different sizes and levels of courses, and coefficients of space required for each level of personnel.

3.3 Reference Data Elements

Sometimes we need data elements for purposes of reference. Consider, for example, a transaction regarding the purchase of supplies. When the supplies are received the vendor is paid an agreed amount and the transaction is recorded in the financial file. But occasionally we need to trace the transaction and check the invoice. To do this the data system must include some means of reference to the invoice, such as the invoice number. This means of reference is a data element.

Other examples of reference data elements are admissions form numbers, grade sheet numbers, purchase order numbers, etc.

3.4 Identification Data Elements

Transactional data needs to be identified as to where and when the transaction occurred and where the data is stored. For example, courses taken need to be identified by semester and year, and payroll needs to be identified by week, month or year.

3.5 Linking Data Elements

We discussed the need for linking data element. A social security number of personal identification number was used as an example. Names can also be linking data elements provided they are unique. These could be used to link faculty teaching courses in the course file with personal information on the Personnel file. Similarly, course codes could link students in the student file with courses they take in the course file; facility codes link the space file with the course and Personnel files; and also the organizational unit code could link the finance file with the space and personnel files. These relationships are illustrated in Figure 4.

A linking code is used in institutions that use program classification code in most of the files that then link these files for calculating program resources and costs. Such a code, known as a PCS in the United States, is both important and complex. It will be discussed in detail in Section 6.

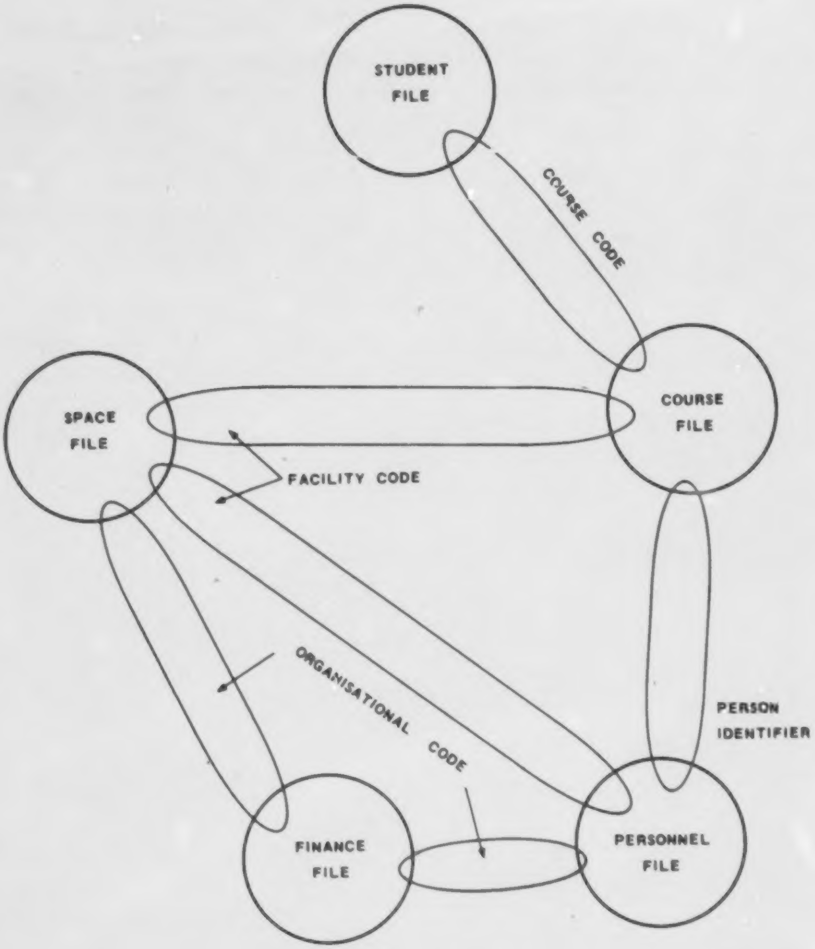
3.6 Checking Data Elements

Many transactions need to be checked for completeness and accuracy. This is sometimes done by matching a total provided as data with a total generated by the computer for the same set (or batch) of transactions. For example, a cashier makes transactions on her cash register and sends these along with a total (or sub-total) generated by the cash register to be processed. If processed at a computer centre the computer totals each transactions in the set and compares the total (or sub-total) with that done by the cash register. If it is the same further processing takes place. Otherwise the inconsistency is traced and resolved. In this case the total (or sub-total) is the value of a checking data element. It typically does not appear in output for a terminal user but is necessary nevertheless.

Sometimes entries of numbers which are long or crucial for other computations need to be checked. Social security numbers (or person identification numbers) are good examples of such a problem. They are often used to assign grades to students or to make payments to employees. They must be recorded accurately when used as input. To prevent input errors a specific additional character of check digit is calculated by a special formula and is attached to each

Figure 4

LINKING ELEMENTS BETWEEN TWO FILES


STUDENT FILE
COURSE CODE
SPACE FILE
COURSE FILE
FACILITY CODE
ORGANISATIONAL CODE
PERSON IDENTIFIER
FINANCE FILE
PERSONNEL FILE

social security number. This check digit is recalculated by the computer using the same formula every time the number is encountered in processing and compared with the original check digit provided with the social security number. Any discrepancy is identified as an error in the input and processing is terminated until the error is corrected.

The check digit attached to the social security number becomes a subfield of the identification number. The social security number then becomes a group data element with the check digit as one data element.

The check digit can be created in many ways. In order for it to be used by more than one user the formula of generation needs to be defined and accepted by users. This subfield will be discussed further in Section 5.2.3.

3.7 Security

To prevent unauthorised access to data, users often need to be assigned codes. One code may be used to identify a users' access to a certain file and another may specify what the user can do with the file (read it or modify it, etc.), and so on. These codes are values of special data elements related to the security of the system.

3.8 Data Elements for Control

Some data elements are needed for the function of control, corresponding to the middle level administrator or manager. Examples of such data elements are average section size, teaching staff teaching load, room utilisation coefficient, variances in budgeted expenditures, and so on.

3.9 Data Elements for Planning

We have mentioned one example of a planning data element, the estimation coefficient in section 2.3. There are many other such data elements when one uses planning models such as CAMPUS, HIS, MSAR, RRPM and TUSS. Examples include faculty loading, student loading, room utilisation, average salaries, and faculty rank distribution.

3.10 Other Data Elements

There are still other data elements that are related to processing. These are primarily of interest to the technical data processor and will not be discussed in this paper.

4. DATA ELEMENT DICTIONARY

4.1 Why a Data Element Dictionary?

We have seen that data is used by all functional levels of administration and management: operations, control and direction, and planning. This is reflected in Figure 5.

Often it is the same data that is used by all levels of administration and may follow organizational lines. This is shown in Figure 6. As an example, consider the value of the data element student enrollment in Mathematics 110. This data is used at the control level to determine if the class should be cancelled or not. If the class is not cancelled then the data is used at the operational level to assign classroom space and other resources for the class. Also, the data on enrollment is used at the planning level for future enrollment projections. The data could also be used to generate derived data like average section size and student-contact-hour (both used for control and planning).

Thus far we have considered the use of the data at the institutional level. It could also be used at the regional or state level. For example, enrollment data and student-contact-hour data are typically used at the state level in funding public institutions. Thus the same data could be used at all levels of decision-making including the national level and perhaps for different purposes at each level. But this common use of data is possible only if all users interpret the data elements in the same manner.

The need for common definitions is not a problem unique to the use of data elements. It arises whenever a large number of people wish to communicate with each other and have a large or complex vocabulary. You then need a dictionary. Most formal languages have dictionaries. Data elements too need dictionaries. One has to be designed for each system using data elements. Such a dictionary used for processing data is called a DED (Data Element Dictionary).

4.2 Designing a DED

The English language (and other natural languages) has many words with more than one meaning and many words that are synonyms. In designing a DED one must try to avoid such linguistic ambiguities. However, this is not always possible since users generally retain their natural language words as the names of data elements. In addition, professionals in the field of data processing have not always agreed on the meaning of technical terms. Only recently has the professional organization of computer scientists in the United

Figure 5
FLOW OF DATA BETWEEN LEVELS OF ADMINISTRATION

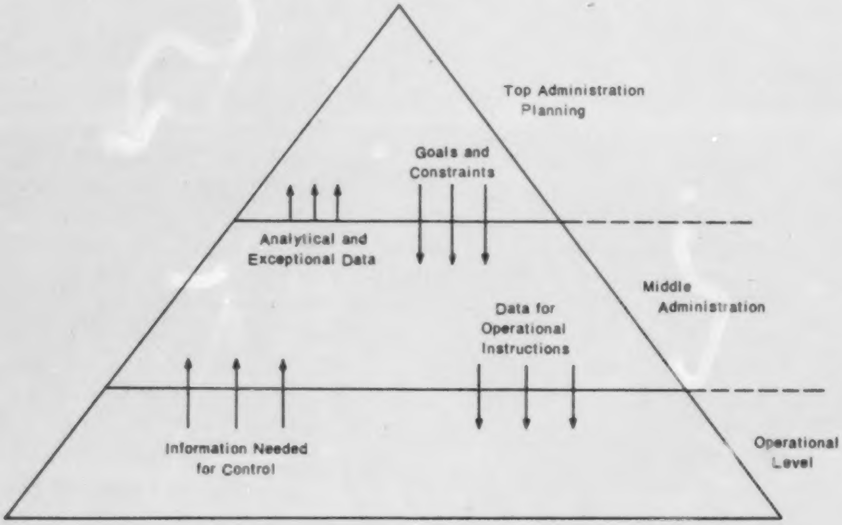
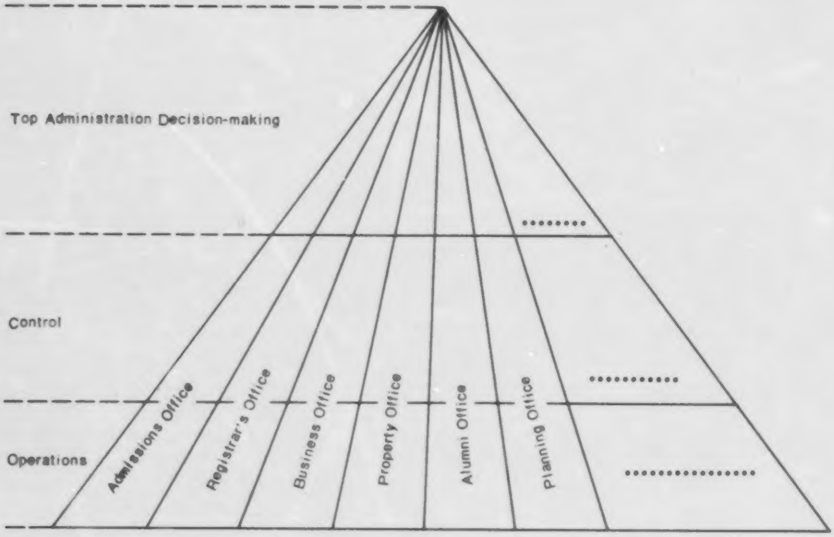

Top Administration Planning
Goals and Constraints
Analytical and Exceptional Data
Middle Administration
Data for Operational Instructions
Information Needed for Control
Operational Level

Figure 6
DATA ALONG ORGANISATIONAL LINES


Top Administration Decision-making
Control
Operations
Admissions Office
Registrar's Office
Business Office
Property Office
Alumni Office
Planning Office

States issued a glossary (ANSI, 1970)(1). But still many practitioners use words with multiple meanings and synonyms. This problem cannot be totally avoided but data element dictionaries carefully designed should minimise ambiguities.

A single dictionary is usually sufficient in the natural languages for defining basic vocabulary. In the case of data elements, however, data elements are organized into a number of dictionaries, each corresponding to a specific file. An institution would therefore have a set of DEDs.

In many countries different DEDs have often been designed at each level of decision making. An educational institution might have one set of DEDs, another set would be at the state (or regional) level, and yet a third set would be for national and federal reporting. This resulted not only in confusion but in a considerable waste of processing resources. This problem can be overcome by having one set of DEDs at the institutional level that is upward compatible. That is, compatible with all DEDs at higher levels of decision making.

Perhaps upward compatibility needs to be elaborated. This can be done by an example. Consider an institutional DED that has one data element to identify the level of a student, that is, the value of the data element identifies a student as being in the first year, second year, third year, or fourth year. The statewide DED has a data element that identifies students as either undergraduates or graduates. Are the institutional and statewide DEDs compatible? The answer is yes, provided that undergraduate is defined in the state DED as being the aggregate of students in the first, second, third and fourth years, and these terms are defined in the same manner as in the institutional DED. Note that the state DED is not downward compatible with the institutional DED because the data element of undergraduate does not disaggregate students by the level of each of the four years.

Data compatibility is desired not only between an institution and its state reporting agency but also between state agencies. This enables information exchange (given other necessary conditions) between state agencies and provides compatible information on states to the national agency. This compatibility is achieved by a DED that defines data elements used by statewide agencies. Such a DED has been developed by the National Centre of Higher Educational Management Systems (NCHEMS) in the United States (Topping & Miyataki, 1972 and NCHEMS, 1974).

---

1) For a glossary on a selected set of terms relevant to computer processing in education, see Hussain (1973: 383-408). For a glossary in prose, see glossary, page 347.

Compatibility of data is also desired between institutions in different states in order to exchange information. This information is mostly at the aggregate level and is related to budgeting and long-range planning. Such information is largely generated by planning related models such as a student flow model, the Cost Finding Principles Project, and a resource requirement prediction model, etc. One prerequisite for compatible output from these models is to have standardized definitions of its input. This is being done in many countries in Europe such as Holland and Germany. In the United States, this was one of the earliest projects of NCHEMS in 1969 resulting in a set of five DEDs currently in its second edition (Goddard et al., 1973).

The DEDs developed by NCHEMS have 215 data elements. They are organized as follows:

1.	Student	70
2.	Course (Classes)	34
3.	Staff (Personnel)	64
4.	Finance	13
5.	Facilities	34
	Total data elements	215

These DEDs correspond to the files typically developed at the institutional level, except that some institutions have an additional file on socio-economic data elements. NCHEMS does not yet have a DED on socio-economic data elements partly because at the time of the development of the DEDs there were no models being developed by NCHEMS that had many socio-economic data elements.

A page from one of the DEDs developed by NCHEMS is shown in Figure 7. Note that this DED not only defines each data element but identifies its uses by different models. One such model identified in Figure 7 is CFP, the Cost Finding Principles Project. This identification of model users of each data element is very valuable when building a data base that will eventually use such models.

4.3 DEDs and a Common Data Base

We have now concluded our discussion of the need and nature of DEDs. We can now discuss the relationship of DEDs with a common data base. This is shown graphically in Figure 8. In this Figure we see the common data base at the centre surrounded by a layer of institutional DEDs. The DEDs are the key to access to the data base for they define the attributes of the data. The institutional DEDs are upward compatible with the statewide DEDs and with data used at the national level.

Figure 7

006 [illegible]

National Center for Higher Education Management Systems

DATA ELEMENT DICTIONARY
Finance Related Data Elements

[illegible] Object Classification

[illegible]

The classification of expenditures according to the nature of the cost incurred.

[illegible] RECORDING INSTRUCTIONS

The Cost Finding Principles Project currently is considering the following categories:

Code	Category	Definition
10	Faculty Salaries	Compensation including fringe benefits to those individuals that the institution considers its faculty (part-time and full-time faculty as well as graduate assistants).
20	Nonfaculty Salaries	Compensation including fringe benefits paid to all employees of the institution except those considered faculty.
30	Supplies and Services	All current operating expenditures other than compensation for personal services (including fringe benefits), expenditures for capital equipment, and stipends.
40	Capital Equipment	Those items of property that have an acquisition cost of $200 or more and an expected service life that exceeds one year.
50	Stipends	Financial assistance awarded to students (both undergraduate and graduate). Includes scholarships, fellowships, and traineeships. Recipients of stipends are not required to render service to the institution as a consideration of their awards, nor are they required to repay them.

[illegible] CFP, IEP, PM, HEFM, RRPM, FAA

COMMENTS

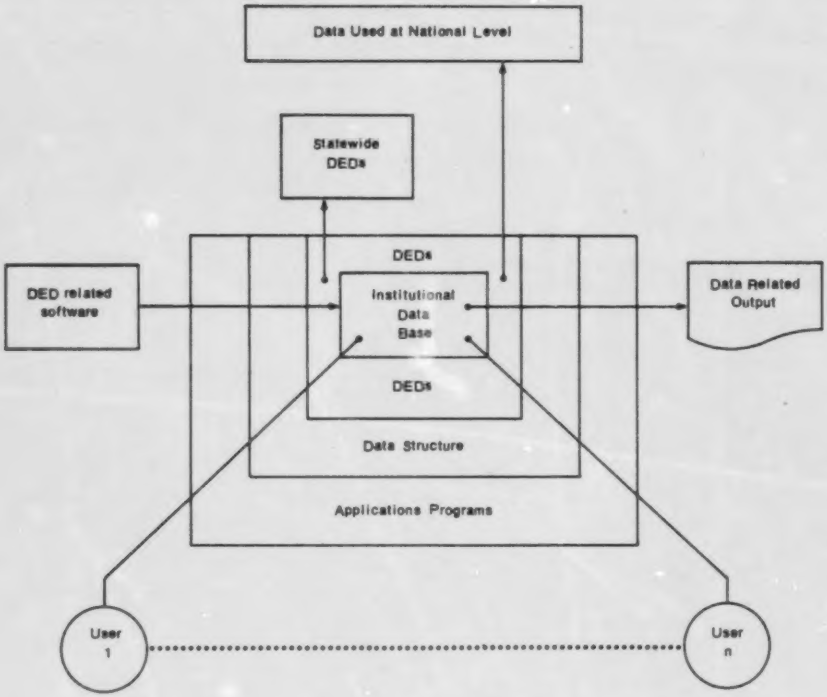
Institutions will usually have more detailed categories within the above groupings.

DATE ISSUED September 1972

62

Figure 8

OVERVIEW OF RELATIONSHIP OF DEDs WITH A COMMON DATA BASE


Data Used at National Level
Statewide DEDs
DEDs
DED related software
Institutional Data Base
Data Related Output
DEDs
Data Structure
Applications Programs
User 1
User n

The DEDs are surrounded by the institutional data structure which defines the manner in which data is stored. The managing of the stored data is facilitated by a set of computer programs which edit, validate, update the data and ensure that it is secure from unauthorised access. These computer programs (software) that are related to the DEDs and the output produced by the DEDs will be discussed in Section 7.

The user application of the stored data is made through an outer layer of computer programs designed for specific institutional applications, such as accounting, student records, payroll, etc.

4.4 Uses of the Common Data Base

We can now restate our concept of a common data base. It is a set of data (values for data elements) that is generalised and integrated for common use. Note that this does not preclude an institution having more than one common data base nor does this imply that all institutional data is in the common data base. One reason that all data is not included in the common data base is that some data is not used by others and is not "common". Another reason that data is not included in the common data base is that users cannot always agree upon what is and should be common data. This raises the question: what data goes into the common data base and why? This question will be discussed in the next section.

The advantages of a common data base are: data required by many users need be collected only once, a single operation will update data for all users, consistent procedures are established for gathering and recording data, and communication between groups (for example, between administrators and analysts) is made possible since one set of DEDs defines terms for all users.

There are disadvantages to a common data base. There is a loss of control by users who formerly owned data. They must now conform to standardized definitions and procedures for assessing and using the common data. Also they lose control over the security and privacy of data(1). We must now answer the questions: what data elements should go into the common data base? To what extent and in what detail should each data element be described? These questions are discussed in the next section.

---

1) There is a distinction between security and privacy of data. Security is concerned with preventing unauthorised access to data. Privacy is concerned with preventing misuse of data. Some of these problems and fears can be overcome by approaches and techniques discussed in Chapter 9.

5. SPECIFICATION OF A DED

The data in the common data base must correspond to the data elements as listed and described in the set of DEDs prepared for the common data base. What are these data elements?

5.1 Selecting Data Elements

Any data element that is common to two or more systems or subsystems (current or planned for the future) must be in the common DED.

Sometimes in an institution a data element may have only one originator, processor or user but it is included in the common data base because it is required by a higher level DED with which the institution wishes to remain upwardly compatible. For example, consider the data element number of books in the library or the fixed cost coefficient for utility costs for the cost centre of physics. In both cases there may be only one institutional user: the resource prediction model. But these data elements are used by more than one institution in a state for purposes of statewide reporting and the data elements are in the statewide DED. Therefore the data elements are "common" and should appear in the common DED.

Factual raw data (i.e. number of books) and estimation data (i.e. fixed cost coefficient) are examples of two types of data that appear in a common data base. Assigned raw data and derived data can also be common data for which data elements need to be defined in the DED. It is sometimes argued that if derived data is based on data elements already in the DED then the derived data need not be included. After all, derived data has no physical reality in the data base, unless specifically stored for later processing. However, derived data needs to be included in the DED for the same reason that other elements are included: that is, to describe the data elements so that no ambiguity arises in communication between originators, processors and users. For a derived set of data, such as the ICLM, who can argue against this need for definition? Even a commonly used data element such as FTE(1) requires a description to avoid misunderstandings. A DED definition of derived data elements also helps the user understand the nature of the data element such as its use and its origin.

How about group data elements? Should they be included in the DED if the component parts are data elements already in the DED? Again the answer is yes. Group data elements need to be defined in the DED for purposes of communication and understanding just as the

---

1) FTE = Full Time Equivalent.

other data elements need to be defined Sometimes even the data elements within a group data element need to be defined for though they look similar to other data elements they may, in fact, be very different. Consider the data element year in the group data element year of birth of a student, for example. In this century a two character data element for year would suffice. But, as we discussed earlier, the data element year in association with the construction of a building on an old campus may require four characters for complete representation. These problems must be anticipated and classified beyond any doubt in the DED.

Linking data elements also need to be in the DED. If files have been integrated using existing data elements the elements will already be in the DED. But if special data elements are created for integrating files, such as social security number, then the element must of course be added to the DED.

How does one determine the data elements that are common to files, reports or users? One determination technique is to use an input output table (I/O table) with the X horizontal axis listing output such as update, files or users and the Y axis listing all data elements. An example of an I/O table is shown in Figure 9. In it we consider output reports and combinations of outputs that correspond to files. The I/O table is of considerable analytic value to both the analyst and the user. It can identify data elements that are used in many files (such as data element No.1 in Figure 9), data elements that are seldom used (data element No.4) and data elements that are never used (data element No.5). The relationship of output (or user) to input (or originator) is graphically portrayed. Data elements that are redundant or can be consolidated with others can be quickly identified.

The preparation of the I/O table for a data base is a simple job for any analyst provided all the required output is known. But this is seldom the case since a data base should have data needed for future as well as current output. This requires predicting needs of future output. This is very difficult for subsystems for which the objectives and policies are not stated. It is also difficult in cases where the manager of the subsystems is not yet on campus. But even when the manager is accessible and the objectives and policies are stated, identifying all the needed outputs is a time-consuming process. All potential users must be involved and a consensus attempted. If a consensus is not possible, then conflicts must be resolved or arbitrated by a higher authority.

In using large models such as CAMPUS, HIS, RRPM, or TUSS, an I/O matrix is necessary to ensure that all the input data elements needed by the model are available in the data base and are in the DED. The data elements can then be checked separately by the analyst

Figure 9

SAMPLE OF I/O TABLE

Reports	File A				File B		
Data Elements	A01	A02	A05	A06	B01	...	...
1. Social Security Number	X	X	X	X	X		
2. Name of Student		X		X			
3. Sex of Student		X			X		
4. Marital Status				X			
5. Hobbies							
6. Units Taken					X		
7. Units Earned		X	X		X		
8. Address	X	X					
9. Date of Birth		X	X	X			

for consistency as to definitions, characteristics and other relevant and desired attributes required by the model. Some data elements of a technical nature need to be in the DED when required by more than one user. For example, data elements on security access and data elements on totals used in auditing fall into this category. But other technical data elements may be excluded when they are common only to the technicians, not to the user. Intermediate data like sub-totals or batch totals are examples of such data elements that may be excluded.

As mentioned earlier, not all data or even all common data will be in the common data base. This may result from disagreements over what is common, or from resistance by former owners of data who do not wish to give up control over their "empire" of data or who are unwilling to permit others to share or know of their data. Also, there may be genuine problems in getting agreement on data element definitions and difficulties in co-ordinating the timing in creating the data. Standardization means each user may have to change practices in collecting and handling data - changes which are naturally resisted even when previous procedures have not worked well. When users resist, the common data base and its DED will evolve slowly but the development should be systematic and progressive with each addition to the DED being economically justified. In practice, however, most decisions on what data is common are political and not technologically or economically rational.

5.2 Description of Data Elements

Once the data elements to be included in the DED are identified, it is necessary to **describe them carefully.** As mentioned earlier, data elements may seem to be the same but be actually different. The data element year in the group data element of date is not necessarily the same as some other data element of year. The data element of address must also be carefully described for there are many different types of addresses, i.e. permanent, local, etc. Thus each data element must be described in enough detail and with sufficient care in the DED so that no ambiguity exists or misinterpretation is possible. This is important not only for accuracy of processing but also for efficiency and effectiveness of processing.

The description of data elements is done by identifying characteristics and attributes which are called descriptors. The primary descriptors of data elements are: name, definition, derivation algorithm, units, format, width of field, validity rules, version of issuance, identification, and status. These will be discussed in detail below.

5.2.1 Name

The name is an identifier of the data element. It must be unique and unambiguous, and, if possible, easy to use and remember. Names must also be logical and meaningful terms.

As for ambiguity, we have mentioned examples of data elements such as address that have many meanings. If an element is named to eliminate all ambiguities, such as home address of father of student, the name may prove too long and cumbersome for convenience and efficiency in processing. As the name length increases more storage and processing time is required. Therefore data element names are often abbreviated. The abbreviation not only economises on processing time and storage but it enables long names to be printed within the limited space of a computer report. If abbreviated for use, then the abbreviation must be defined(1).

5.2.2 Definition

The definition explains the meaning of the data element. Examples of a practical application for the element may be included in

---

1) Names of states in the United States, and names of countries from which students come are abbreviated for national use in the United States. For such lists, see Goddard, et al., (1973: 6.3.1 - 6.4.1).

the definition for added clarity. Other data elements with similar or contrary meanings may be mentioned, and the root of the term identified.

The origin of the data element should also be included in the definition. Thus if a data element is raw data (a fact or observation), its origin identification should be included to give a better understanding of the data element. If the data element is estimated, the manner of estimation is relevant.

In defining a data element it is important to define it at the lowest level of aggregation in case this detail is needed later on. We have already discussed the example of defining the student level at the first year, etc., even though the aggregate level of undergraduate may be all that is required. One must weigh, however, possible benefits from lower levels of aggregation against the cost of including non-essential data elements in the DED and the added cost of processing these data elements.

5.2.3 Derivation Algorithm

If data is derived then the algorithm for its derivation must be specified. This not only eliminates confusion or misunderstandings in the definition but it is essential for the programmer who will have to write program instructions to derive the data.

The importance of specifying the derivation algorithm can be illustrated by an example. Consider the data element FTE. In the United States, the FTE is the student-credit-hour or student-contact-hour value divided by a full time student load. But which should be used - student-credit-hour or student-contact-hour? And what is a full time student load? Is it 11, 12, or 15 credit or contact-hours (a very important distinction in some countries)? The number varies between states and within a state it may vary for students at the undergraduate and graduate levels. The exact value chosen for the calculation must be clearly specified. There is a further ambiguity in calculating the FTE. It concerns the student who is included in this calculation. Does the term student include auditors? Does it mean students that take remedial non-credit courses? This distinction can greatly affect the calculated result, and therefore, the type of student in the derivation algorithm must be carefully specified.

In deriving values it is necessary to specify the level of accuracy required. For example, should FTE be calculated to one, two or three decimal places and how should it be rounded? Rounding becomes especially important in calculating financial values. A rounding (unauthorised) algorithm led to the classic salami payroll case that accumulated the rounded money from each paycheck into the programmer's account.

One must also be careful to specify exactly which algorithm is used when many exist for the same purpose. For example, two common techniques exist for calculating the check digit: Mod 10 and Mod 11. But there are numerous others used(1). When many algorithms exist for the same purpose or many could be formulated, the derivation algorithm must specify exactly what formulation is used and why.

Problems of derivation also occur with alphabetic data. For example, consider the data element name of student. Most computer outputs limit name of student to 20 characters, but many foreign students have names much longer than 20 characters. In Portugal, 176 characters for name are allowed by the government on some of its application forms. How should institutions in other countries and especially in Portugal abbreviate a 176 character name? Let the student choose an abbreviation or develop guidelines applicable to all names? What abbreviation guidelines should be selected? Initials? Truncations? The abbreviation rules used for data elements must be explicitly defined.

As another example, consider a course such as The Modern History of Mexican Art from 1952-1974 which must be abbreviated for printing in a course schedule. The prepositions in the name may be deleted but little else. The remaining words can be abbreviated but this must be done according to agreed-upon conventions. This author once abbreviated English 101 to Eng. 101 in a course schedule with disastrous results. Students wanting to register for Engineering 101 registered for English 101 by mistake. Thus abbreviations must be carefully chosen. Abbreviated course names create a new data element, abbreviated course title, as distinct from course title which is the full unabbreviated title. When acronyms are used: (e.g. HIS, FTE, CERI, OECD), they too should be named and defined.

5.2.4 Units

The units of each data element must be specified. This can be crucial in some calculations. As an example, consider ICLM, the Induced Course Load Matrix. It identifies the course load generated by students of each major in each instructional program at each student level. The sum of the load for all the students for any one course or activity would then give the total course load which when divided by the average teaching staff load gives the number of teaching staff required. This is the "heart" of all resource prediction models like CAMPUS, HIS, MSAR, RRPM and TUSS. But note that we have used the term load twice: once in the ICLM and once as the teaching

---

1) For a discussion of nine algorithms, see Daniels and Yeates (1971: 70-71).

effort. This load could be expressed either in the units of contact-hours or units of credit-hours or hours of work assignment. Whichever unit value is chosen, load must be consistently used in the calculation. This example shows the importance of carefully specifying units of data elements in the DED.

5.2.5 Format

The format of how the data elements are stated is also a descriptor to be included in the DED. This arises in group data elements. We mentioned the date as an example in an earlier discussion. Usually the format for date is:

MM
└month

DD
└day

YY
└year identification by last
two digits

If this is the desired format it should be so stated. Otherwise a possible misunderstanding might arise for DD MM YY is a format used in England. Another format commonly used in some European countries is YY MM DD.

The format of a name is also important. a DED that defines a student's name as a combination of surname, first name and middle initial does not have an adequate definition. The format of the name must also be specified. This includes the specification of the order of the name components and the separators between the component parts. A blank as a separator can be confusing because it sometimes occurs within a surname such as Mac Donald or Van der Kamp. Suffixes like Jr. and Sr. must also be specified. There is also the problem of shortened names (Pat for Patrick or Bob for Robert) and nicknames. Are they allowed? This is not hairsplitting. The detailed specification is necessary or names will not match in computer processing. Names must have exactly the same format whenever they appear, if they are for marking. That is, the same order of components, the same separators, and the exact same spelling.

Another formatting problem that must be specified in the DED is the placement of data in a field wider than the data itself. Should the data start from the left or the right of the field? In the case of numeric data it is customary to start from the right, called right justified. Otherwise the resulting blanks could be misinterpreted as zeros, and add tens, hundreds or even thousands to the value of the number. In the case of alphabetic data one customarily starts from the left, called left justified, if one expects to sort the data and also to facilitate reading a list of alphabetic data. This left or right justification, if used, must be so specified in the DED.

5.2.6 Width of Field

The DED definition should also specify the width of the field for each data element. We discussed this problem earlier when we discussed the data element year in the group data element of date. Most width determinations are technical. The descriptor can be provided by the analyst. For example, adding a check digit to a social security number is a technical decision an analyst must make. But some widths cannot be determined by the analyst. They require an administrator's projection. For example, how many characters should be allowed for the data element student enrollment in institution? An amount of expenditure of 2,000 would require a 4 character field width. But if the amount increased to 10,000 or more then the field would not have enough space. (An amount of 12,002 would appear as 2,002 on some computer printouts when the field has only 4 characters, ignoring the most significant digit.) The potential extreme values must often be projected, as in this case. This is the responsibility of the administrator, not the analyst. This is not a technical problem though the technician is involved.

Administrators must also define widths that involve value judgements. As an example, consider the width of surnames of students or staff. The technician can determine statistically that 70 per cent of the surnames are 10 characters or less, 82 per cent are 12 characters or less, and so on. Using these statistics he can state the probability of a person's name being truncated because of insufficient width. But the psychic loss of truncation the administrator alone can state. He must weight this loss against the use of space caused by long names in reports and, of course, the cost of processing long names. The ultimate decision in such cases should rest with the administrator, not with the analyst or technician.

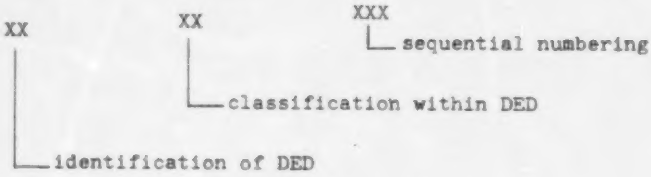
5.2.7 Validity Rules

Many data elements in the DED have values that need to be checked for validity. These data elements should be identified and the validity check rules stated in the DED. The analyst can specify many of these rules. For example, he can state that a name field must not have numeric characters and he can specify the use of special symbols like hyphens. Similarly, he can specify data element that must be numeric, like number identifications of students. Some check rules, however, must be determined by administrators rather than analysts. These include checks as to probable ranges of data element values. For example, in a university a check rule for age of student could identify values under 10 and over 85 as definite errors and ages between 10 - 15 and 75 - 85 years as possible errors. The administrator can best state these ranges. He knows the organization best and knows what data needs to be checked for allowable ranges.

He also knows what data is essential and cannot be missing from the data file. All these validity rules must be stated in the DED.

5.2.8 Identification

Each data element should be labelled so that it can be easily and quickly identified. The name of the data element might be used for this purpose but names are not conclusive to unambiguous quick reference. The alternative is to assign each data element with a special identification number. The numbering could be sequential within each DED. This provides uniqueness and unambiguity but not necessarily ease in retrieval for reference in cases of large files. A blocking decimal code structure has been successfully used for this purpose. An example is:



The first character or two characters identifies in which dictionary within the set of DEDs the data element appears. The next subfield identifies how the data element is classified within the DED. For example, in a payroll file a subclassification might be the rate of pay, the tax deduction rate, the medical service deduction rate, etc. The third subfield can be a sequential identification. An example of this coding structure is shown in Figure 10.

Figure 10

EXAMPLE OF CODING FOR DATA ELEMENT IDENTIFICATION

DED Identification	Classification of Data Elements	Sequencing of Data Elements
03 Staff	01 Name	01 Surname 02 First Name 03 Middle Name 04 Jr. or Sr. Identification
	06 Payroll	01 Rate of Pay 02 Health Deduction Rate 03 Tax Deduction Rate

There may be more subfields than shown in Figure 10 if greater sub-classification is desired. For example, one might establish another subfield under Health Deduction Rate (03.06.02 in Figure 10) to identify deduction rates for married students and unmarried students If 01 identified the unmarried student rate, and 02 the married student rate, then the identification of the data element for an unmarried student's health deduction rate would be 03.06.02.0.1.

The main purpose of subfields is to facilitate reference, processing and analysis. Once the relevant subfields are identified the remaining subfields can be ignored. For example, in searching for data elements concerning staff payroll (using Figure 10), only the codes 03.06 need to be identified in the first two subfields. This lumping and clustering by sub-classification also helps efficient maintenance since it quickly identifies data elements that are duplicates or close duplicates needing consolidation.

5.2.9 Status

The status of data elements should be specified. A data element is either operational or under consideration. The latter status may be further specified as to whether it has been merely proposed or whether it has been approved, or incurred, but is not yet implemented. Finally, there is the status of being superseded but still recorded for reference purposes.

5.2.10 Other Descriptors

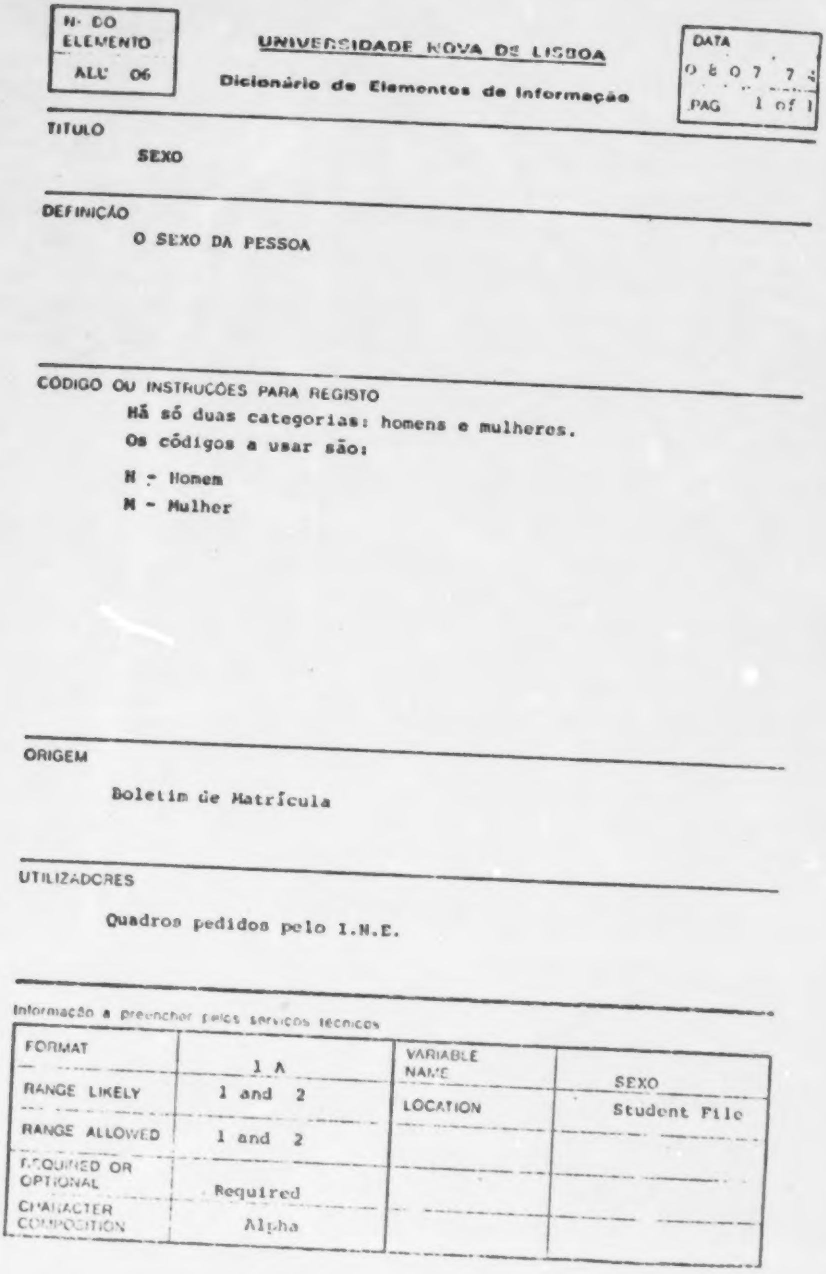
Thus far we have limited our discussion to basic descriptors. Other descriptors are necessary if the DED is to be used for technical and analytical purposes. Some of these will be discussed in Section 7.

5.3 Content of the DED

The descriptors discussed in the preceding pages should be used to specify each data element in the DED. However, the ordering of descriptors used in this text need not be the ordering used in the DED. The DED order is generally dictated by constraints of space, by the importance of descriptors, by ease of reference, and finally, by logical groupings of descriptors such as the groupings of technical descriptors. An example of all the contents of a DED page is shown in Figure 11. This figure should be compared with Figure 7 which shows another DED page. The DED in Figure 7 is more concerned with definitions and uses of the data elements than is the DED in Figure 11. The latter (Figure 11) is more concerned with the creation and maintenance of the data base. Consequently, it has more technical

Figure 11

SAMPLE OF DED FORM



N· DO ELEMENTO
ALU 06

UNIVERSIDADE NOVA DE LISBOA

Dicionário de Elementos de Informação

DATA
0 8 0 7 7

PAG 1 of 1

TITULO

SEXO

DEFINIÇÃO

O SEXO DA PESSOA

CODIGO OU INSTRUCÕES PARA REGISTO

Há só duas categorias: homens e mulheres.

Os códigos a usar são:

H - Homem

M - Mulher

ORIGEM

Boletim de Matrícula

UTILIZADORES

Quadros pedidos pelo I.N.E.

Informação a preencher pelos serviços técnicos

FORMAT	1 A	VARIABLE NAME	SEXO
RANGE LIKELY	1 and 2	LOCATION	Student File
RANGE ALLOWED	1 and 2		
REQUIRED OR OPTIONAL	Required		
CHARACTER COMPOSITION	Alpha		

descriptors. These technical descriptors can be glanced at or skipped by the administrator. The rest of the data descriptors, however, are used by the administrator. These descriptors (except for identification codes and dates) must be determined by the administrator. This is in his interest since he is the ultimate user. Determining descriptors is his responsibility. If the administrator assigns this responsibility to the analyst or shifts the responsibility to the analyst by default there is the danger that the analyst will not perceive the problem or environment in its entirety as would the administrator. The analyst lacks the background, knowledge and experience with the organization required for the task. Determining descriptors is a function for administrators in the development of DEDs and the data base.

6. PROGRAM CLASSIFICATION CODE

6.1 Introduction

One important element in any data base in higher education is a Program Classification Code (PCC). It identifies uniquely every program and subprogram in higher education. In many institutions it is used as a basis for budgeting (when PPBS - Program Planning Budgeting System - is used) and long range planning. In some countries it is used for national reporting.

The Program Classification Code reflects the structure of the educational system and therefore varies between countries with different educational philosophies and structures. Nonetheless, it is instructive to examine at least one such classification system in order to appreciate its complexity and the issues involved in its implementation. For such a discussion, the system chosen is the Program Classification Structure (PCS) used extensively in the United States. Its construction, structure and format will be examined, followed by a general discussion of the use of the Program Classification Code (PCC).

6.2 The PCS

6.2.1 Construction of the PCS

The PCS was developed by the National Centre of Higher Education Management Systems (NCHEMS). As with all the products of NCHEMS, the design was done by a national task force selected from institutions of higher education and tested by a stratified sample of pilot institutions for validity and generality. It was also tested by pilot institutions implementing other NCHEMS products such as the RRPM

(Resource Requirement Prediction Model), the CFP (Cost Finding Principles Project), etc. Comments and critics were solicited from the over 2,500 institutions of higher learning in the United States and were reviewed regionally and nationally by all interested groups such as statewide and national agencies as well as academic professional organizations. The modified PCS, after three years of development, was released in 1972 (Gulko, 1972). It is currently being revised for special problems including those raised by vocational and trade institutions.

6.2.2 The Structure of the PCS

The PCS classifies programs into one of three primary programs or one of four support programs. Each has subprograms with a total of 30 subprograms. These are shown in Figure 12 (Gulko, 1972: 19).

The PCS has six levels of classification. These are shown in Figure 13 (Gulko, 1972: 21). The lowest is a program element which is then associated with descriptors called program measures which enable each program element (and its aggregations) to be related to the "resource utilised, activities level generated, groups served, expenditures incurred, revenues generated and outcomes achieved" (Topping, 1972). This data provides us with a production function for each program and enables program planning and program budgeting.

6.2.3 Code Format of the PCS

The PCS in its code format is shown in Figure 14 (Gulko, 1972: 26). The program categories and subcategories of the PCS incorporate a classification long used by the national government for annual reporting by all institutions of higher education. This classification, known as HEGIS (Higher Education General Information Survey), was revised by yet another project of NCHEMS (HEGIS VIII) and subsumed in the PCS.

The PCS is also incorporated or adopted by many statewide agencies. In such states, an institution adopting the PCS needs only one set of program code for institutional, statewide, and national reporting. This eliminates the need of three sets of codes required formerly and not only adds to efficiency in processing but also reduces errors and confusion resulting from different sets of codes.

6.3 Use of the PCC

The most important use of a PCC is to enable program budgeting and analysis. As stated by Gulko: "The desire to apply programmatic analysis to budgeting and planning in higher education and the growing need for accountability has identified the need for a structure that will provide a consistent and organized means of classifying all activities of higher education in terms of programs". (Gulko, 1972).

Figure 12

ORGANISATION OF THE PROGRAM CLASSIFICATION STRUCTURE

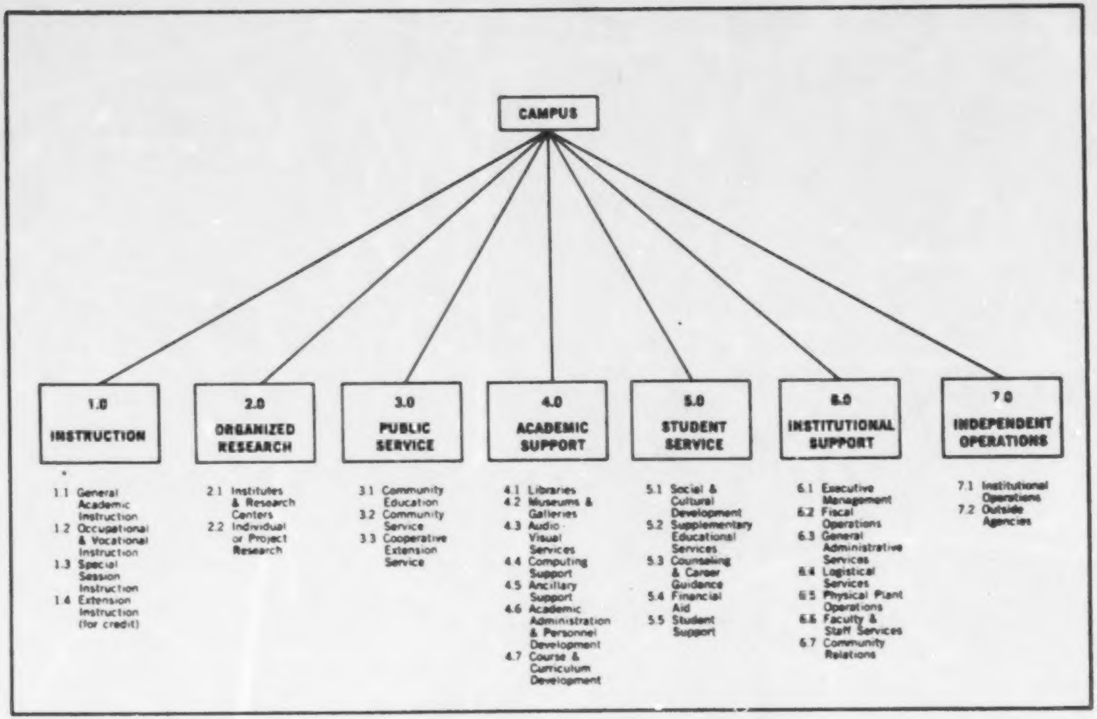

CAMPUS
1.0 INSTRUCTION
2.0 ORGANIZED RESEARCH
3.0 PUBLIC SERVICE
4.0 ACADEMIC SUPPORT
5.0 STUDENT SERVICE
6.0 INSTITUTIONAL SUPPORT
7.0 INDEPENDENT OPERATIONS
1.1 General Academic Instruction
1.2 Occupational & Vocational Instruction
1.3 Special Session Instruction
1.4 Extension Instruction (for credit)
2.1 Institutes & Research Centers
2.2 Individual or Project Research
3.1 Community Education
3.2 Community Service
3.3 Cooperative Extension Service
4.1 Libraries
4.2 Museums & Galleries
4.3 Audio-Visual Services
4.4 Computing Support
4.5 Ancillary Support
4.6 Academic Administration & Personnel Development
4.7 Course & Curriculum Development
5.1 Social & Cultural Development
5.2 Supplementary Educational Services
5.3 Counseling & Career Guidance
5.4 Financial Aid
5.5 Student Support
6.1 Executive Management
6.2 Fiscal Operations
6.3 General Administrative Services
6.4 Logistical Services
6.5 Physical Plant Operations
6.6 Faculty & Staff Services
6.7 Community Relations
7.1 Institutional Operations
7.2 Outside Agencies

Figure 13
PROGRAM CLASSIFICATION STRUCTURE NOMENCLATURE

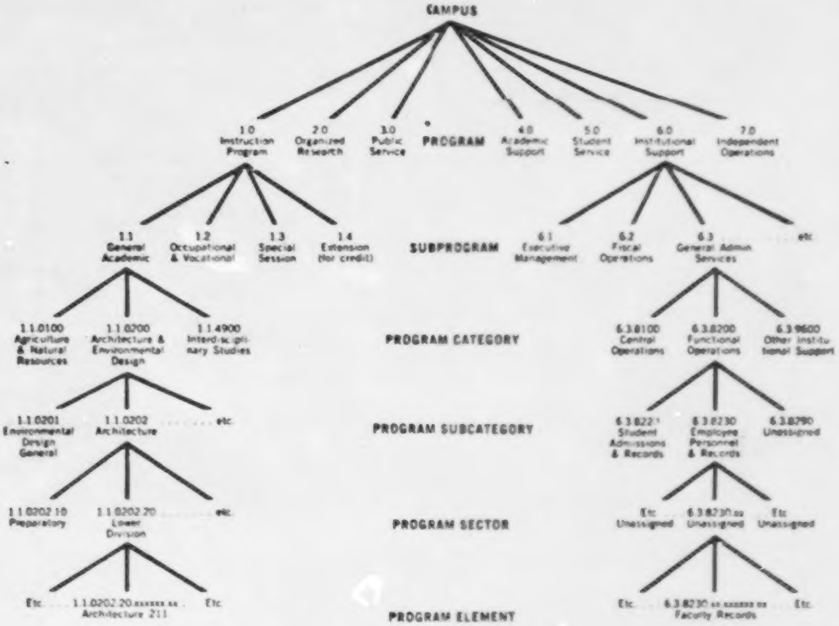
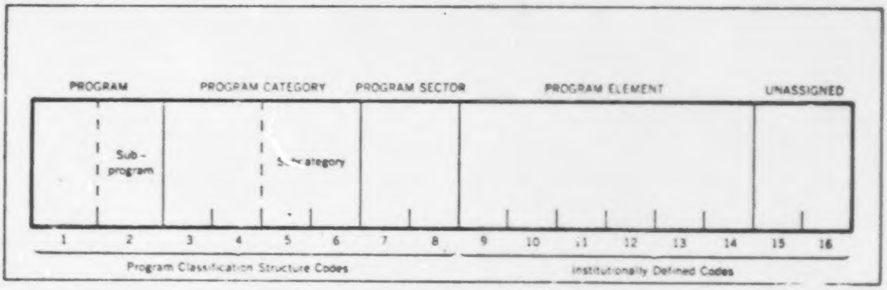

CAMPUS
1.0 Instruction Program
2.0 Organized Research
3.0 Public Service
PROGRAM
4.0 Academic Support
5.0 Student Service
6.0 Institutional Support
7.0 Independent Operations
1.1 General Academic
1.2 Occupational & Vocational
1.3 Special Session
1.4 Extension (for credit)
SUBPROGRAM
6.1 Executive Management
6.2 Fiscal Operations
6.3 General Admin Services
etc.
1.1.0100 Agriculture & Natural Resources
1.1.0200 Architecture & Environmental Design
1.1.4900 Interdisciplinary Studies
PROGRAM CATEGORY
6.3.8100 Central Operations
6.3.8200 Functional Operations
6.3.9600 Other Institutional Support
1.1.0201 Environmental Design General
1.1.0202 Architecture
etc.
PROGRAM SUBCATEGORY
6.3.8220 Student Admissions & Records
6.3.8230 Employee Personnel & Records
6.3.8290 Unassigned
1.1.0202.10 Preparatory
1.1.0202.20 Lower Division
etc.
PROGRAM SECTOR
Etc. Unassigned
6.3.8230.xx Unassigned
Etc. Unassigned
Etc.
1.1.0202.20.xxxxxx.xx Architecture 211
Etc.
PROGRAM ELEMENT
Etc.
6.3.8230.xx.xxxxxx.xx Faculty Records
Etc.

Figure 14
CODING STRUCTURE FOR THE PCS


PROGRAM
PROGRAM CATEGORY
PROGRAM SECTOR
PROGRAM ELEMENT
UNASSIGNED
Sub-program
Subcategory
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Program Classification Structure Codes
Institutionally Defined Codes

The PCC also provides the program dimension to many planning related models such as the Teacher Activity Analysis, Manpower Accounting, Budgeting Manual, a resource allocation model, and costing models. The results of some of these models, especially the costing and allocation models, then enable the exchange of information on cost and performance.

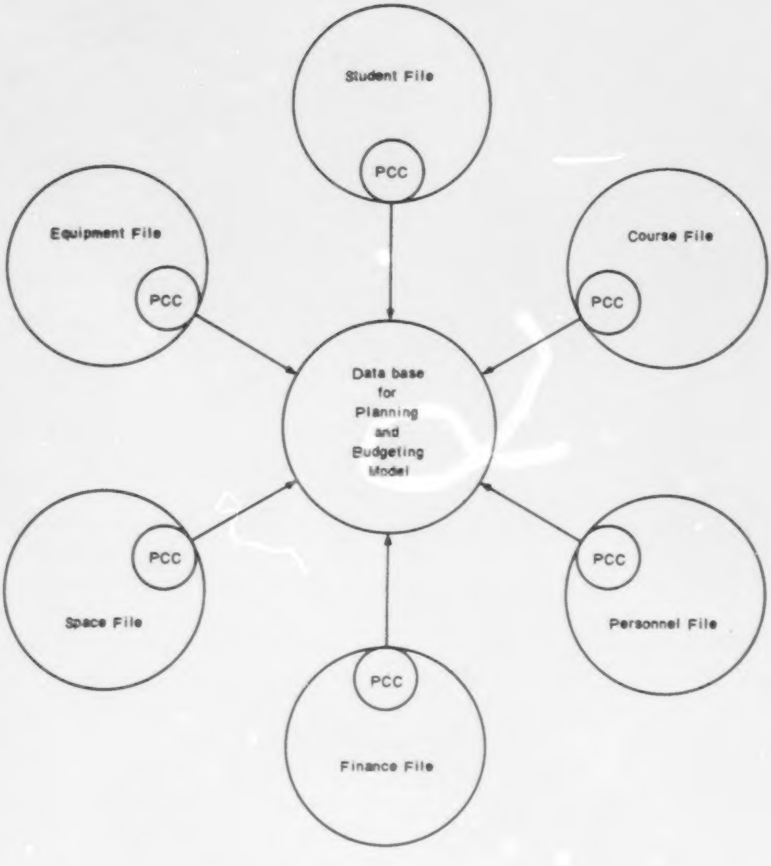
The PCC structure should not be construed as a new chart of accounts for an institution nor as a replacement for existing data coding systems. It is, however, necessary to map data elements from the institutional data system to the PCC or alternatively to incorporate the PCC as a data element in the institutional data system.

The PCC could then be used to relate the many relevant data elements to each program or subprogram. In a budgeting and planning system, this would require that the PCC be in every file needed for programmatic analysis.

An illustration may help. Consider that the institution wishes to calculate the direct cost per student contact-hour for each program as classified in the PCC for purposes of statewide reporting or national cost exchange. To do this the payroll or staff file, the space file, the finance file and other relevant files must be searched for direct costs associated with each PCC (program or subprogram). By adding cost components one gets the total direct cost for each PCC. By next going to the student and/or course file, one can calculate the student contact-hour for each PCS program. Dividing the direct costs by student contact-hours for each corresponding program one gets the direct cost per student contact-hour for each program. Note that in order to make this calculation we needed the PCC in each file. In other words the PCC performs the function of a linking element. Such a role is shown for the PCC in Figure 15, assuming a typical set of functional files. This role is relevant not only for information exchange, but for a data base for use of models like CAMPUS and RRPM.

Having the PCS in each functional file requires that the administrator associate resources like personnel, space, equipment, etc., with a PCC. In the case of some resources, like space and equipment, this may be a one shot task. But in the case of personnel, revisions must be made every year in case personnel assignments change. The task of assigning resources to the PCC is particularly difficult when personnel work for more than one PCC; that is, when some personnel work in both instructional and support programs, or when they have both teaching and administrative duties within an instructional program. In each case, the proportional distribution of load for each PCS must be assigned. These assignment values could be calculated from Faculty or Staff Management Analysis Forms. But if these are not used or are not reliable, the assignment values could come

Figure 15
PCC AS A LINKING ELEMENT


Student File
PCC
Equipment File
PCC
Course File
PCC
Data base for Planning and Budgeting Model
PCC
Space File
PCC
Personnel File
PCC
Finance File

from the budget or be assigned directly by an administrator. In any event, the administrator is involved. But in order to perform this task he must understand the nature and structure of the PCC system. He must also appreciate the significance of the assignments to each PCC because that determines the costs and loading by program. The administrator must not abdicate this task or let it go by default to the analyst. This is an important input, especially for a resource prediction model like RRPM and CAMPUS (other models like HIS, MSAR and TUSS do not as yet do costing). It must reflect reality as much as possible, or the output results will not be accurate. The output is only as good as the input that is processed. And the administrator is responsible for the input. He is, of course, the beneficiary of its results.

IMPLEMENTATION

7. USES OF A DICTIONARY/DIRECTORY SYSTEM

7.1 Introduction

Directories are lists or tables that facilitate quick reference to pertinent information. Consider a directory in everyday life - a telephone directory. We refer to it for information regarding a person's address and phone number. The key word or index in this case is the person's name. In data systems we also have directories. These index data elements and provide information about the elements such as where data elements are located in the files, who originated and who uses the elements, and which programs and transactions use the elements. In most directories the key word indexed is the data element name.

Since DEDs also provide information about data elements (definitions), dictionaries and directories are closely related. In some cases directories also define data elements so they perform the function of dictionaries. The distinction between directories and dictionaries has been further confused by many data processing practitioners who define fixed data elements such as sex in dictionaries and variable data elements such as marital status in directories. This distinction causes confusion since not all data elements can be categorized in this manner. Some data elements are fixed over a period of time and then change, such as a girl's name which changes upon marriage. Also data element definitions often change when managers change. There simply is no universally accepted distinction between dictionaries and directories in the data processing profession(1) so it is customary to refer to the two together as a Data Element Dictionary Directory System, or simply a DED/DD system.

Directories are used extensively in Data Base Management Systems. Since such systems are beyond the scope of this paper only a passing reference to such uses will be made. Instead we shall discuss directories that are implemented outside a Data Base Management System(2), and their use in conjunction with dictionaries.

The DED/DD system can be classified conceptually into two parts. One part is used by people (both data processing and non-data processing personnel) and the other by computer software (both applications programs and computer system programs). The latter is rather technical and will not be discussed in any detail. The former does concern the administrator and will be discussed at some length.

7.2 Uses of Administrator and Other Personnel of the DED/DD System

7.2.1 Use as a Repository of Information

One important use of the DED/DD system is that it documents information related to the processing of data for data which is common to two or more current or future organizational units. This documentation repository also serves as a set of standards governing the creation and use of the data.

Accessing this information is difficult because of the complexity and massiveness of the DED/DD system. In large DEDs and DEDs having long or compounded names, it is sometimes difficult to identify data elements, especially when one has forgotten the exact name or spelling of the element. An index is useful in such cases, such as the KWIC (Key Word in Context) or KWOC (Key Word Out of Context). The KWIC index is the more commonly used because it identifies in alphabetic sequence every word without changing the order of the words. An example of this is shown in Figure 16 which is an index for H words of a sample KWIC index, prepared for the data elements of all the five DEDs developed by NCHEMS. In it, all data elements that have a main word starting with an H (shaded area) are indexed alphabetically and are referenced (by DED number and page number) irrespective of the position of the word (first, middle or last).

---

1) For a discussion of dictionary and directory definitions see Canning (1974a: 12), 12 pages. This is a review of Data Base Management and concludes that dictionary and directory are terms still ambiguously defined.

2) For examples of such dictionary/directory software see Canning (1974b). One such system specifically for higher education is CASCADE, developed by the Wisconsin State University College at Oneonta, Wisconsin, U.S.A.

Figure 16
EXAMPLE OF A KWIC INDEX

	Name of Data Element		DED Number	Page Number
	Handicap	Status	4018	4.3.20
	Highest	Degrees/Diplomas Earned	4029	4.3.31
			5023	5.3.25
	Honorary	Degrees and Awards	4023	4.3.25
Course Contact	Hours		1012	1.3.13
Faculty Contact	Hours		1208	1.5.9
Number of	Hours	Worked	5310	5.6.11
Section Contact	Hours		1106	1.4.7
	Housing	Status	5011	5.3.13

Figure 17
SAMPLE TABLE SHOWING POTENTIAL AND FUTURE USES OF DATA ELEMENTS

Number	Title	Description	Comment	Uses HEGIS	Level III Cost Exchange	Leve Mode
001	Student name	Legal name of student				
002	Student Number	A number to uniquely identify each student				
003	Social Security Number		May serve as a student number (preferred)			F
004	Sex			X	X	[illegible]

Key: F = Likely use by future WICHE simulation model. X = Use by WICHE simulation model.

Similarly, any data element can be referenced by indexing alphabetically any one of the main words in its name. For example, consider the data element number of hours worked. It can be indexed alphabetically in the N's under number, in the H's under hours (as in Figure 16), and in the W's under worked, but not under "of", which is a preposition and not a main word. The KWIC index thus enables quick referencing and is necessary with long lists of names. The KWIC index technique also has many applications in educational management other than indexing names of data elements - such as preparing bibliography lists for libraries, or lists of chart of accounts.

Some data element names are ambiguous and cannot be identified uniquely. For such data elements and for those whose names are chosen for convenience rather than relevance, another descriptor, called a designator, is added to the definition of the data element. This descriptor is a set of key words reflecting the content and meaning of the data element. These key words are then also used for indexing the data elements.

For example, consider the data element name of student when, due to excessive length, the name has been shortened or reduced to 20 characters. The data element requires a designator identifying the key words shortened, name and student. The KWIC would list this data element under each of these three key words. The designator can be made more readable by adding prepositions such as "of", "which is" and "which are" to the key words (i.e. shortened name of student). The most common of these prepositions is "of" which is why this capability of adding prepositions is referred to as the OF language (IBM, Form SH20-1105 and Uhrowczik, 1973: 349).

The designator of data elements lists key words in a prescribed order so that the relative generalities of the key words can be identified. The IBM convention (Uhrowczik, 1973: 348) starts with the most general key word and proceeds to the most specific. Using the IBM convention for the shortened name of a student, the key words must appear as name, student and shortened, since name is the most general key word and shortened is the most specific.

Some key words occur frequently. These are designated classwords and must be used wherever applicable. This helps reduce the use of synonyms. Examples of classwords are: amount, code course, date, faculty and number.

Using the designators for indexing, and using key words in designators enables the grouping of similar data elements. This greatly facilitates the identification of possible duplications, synonyms, and inconsistencies. Such an analysis can greatly contribute to the construction and maintenance of an efficient and easy way to use DED.

7.2.2 Identifying Uses and Users of Data Elements

The relationship of each data element with its possible use can be represented as a table or matrix. In it data elements are represented on one axis and the use (current and potential) on the other axis. An entry in a cell identifies the user of particular data elements.

An example of such a matrix is shown in Figure 17. It is part of the DED published by NCHEMS (National Centre for Higher Education at Western Interstate Commission for Higher Education)(Thomas, 1970). The use columns represent models and the cells within the use columns identify the use of each data element by each model. Such a table in a DED is appropriate for a national agency or a statewide agency that provides its constituents (institutions of higher education) information on what data elements are required for each model or application. Potential users can refer to such a table to determine what data they need for different models and applications.

A similar matrix was the I/O matrix shown in Section 5.1. In it, however, the columns identify the reports that actually use different data elements. A variation of this matrix would be to have the columns representing organizational units rather than reports. Such a matrix is shown in Figure 18. From it the analyst knows what organization unit is affected for each data element.

7.2.3 Identifying Data Creators

Another matrix that is useful is one that identifies organizational units that create and/or maintain each data element. The matrix thus specifies organizational unit responsibility for the collection of data elements. This responsibility includes that the data be correct, complete and collected in time.

Such a matrix can be especially useful when superimposed on the matrix showing organizational units that use each data element. Such a matrix is shown in Figure 19.

Examining Figure 19 one finds that some data elements are created and used by the same organizational unit. The organizational unit therefore has a vested interest in the quality of the data. Other users of the same data can be assured of the quality in most cases assuming that the creator of the data is motivated and conscientious. But there are some occasions when one organizational unit collects the data but does not use it. The collecting unit may then become lax in the quality of the data generated. This danger should be identified and measures adopted to motivate organizational units to collect accurate data. In addition, all data generation should be monitored.

Figure 18

SAMPLE TABLE SHOWING ORGANIZATIONAL UNITS AS USERS OF DATA ELEMENTS

Data Elements	Users Organizational Unit						
	Admissions Office	Registrars Office	Academic Deans	Academic Department Heads	Dean of Students	...	...
For Students							
1. Name	X	X	X	X	X		
2. I.D. Number	X	X	X	X	X		
3. Sex	X	X			X		
4. Marital Status		X			X		
5. Residence		X			X		
6. Major	X	X	X	X			
7. Level		X	X	X			
8. Courses Taken		X	X	X			
9. Grades for Course		X	X	X	X		
10. Grade Point Average		X	X	X	X		

7.2.4 Identifying Transactions

It is sometimes desirable to identify the data elements used by each transaction. A sample of such a matrix is shown in Figure 20. In the matrix transactions can be either formal input documents and printed forms or formal output forms. The matrix shows which transactions would be affected if a data element were altered (i.e. deleted or redefined). Such information is useful for form and document control.

7.2.5 Descriptors Needed for 7.2.1-7.2.4

In Figures 18-20 the data elements appear in the cross reference tables by name. In practice they are often referenced by an identifier called a label which is chosen to be unique, short and easy to process.

Figure 19

SAMPLE TABLE SHOWING CREATORS OF DATA ELEMENTS

	Organizational Unit						
Data Elements	Admissions Office	Registrars Office	Academic Dean	Academic Department Heads	Dean of Students	...	...
For Students							
1. Name	▨	X	X	X	X		
2. I.D. Number	▨	X	X	X	X		
3. Sex	O	X			X		
4. Marital Status		O			X		
5. Residence		O			X		
6. Major	▨	▨	▨	X			
7. Level		▨	X	X			
8. Courses Taken		▨	▨	▨			
9. Grades for Courses		X	X	▨	X		
10. Grade Point Average		▨	▨	X	X		

Key: X = User
O = Creator
▨ = Creator and User

For example, the data element grade point average can be identified by the label GPA. This label could also be used to identify the data element in computer programs. Some programming languages do have limitations as to the choice of labels. For example, the FORTRAN language has a restriction on the number of characters allowed for identification labels. So labels must be designed with the projected needs of language taken into consideration.

The cross reference tables and indexes discussed earlier in this chapter can be generated by computer programs. In most cases this requires a trivial effort(1). But there is an important cost

---

1) The effort of programming the KWIC/KWOC Indexes is not trivial in itself but the indexes are packaged programs and are commonly available for this application.

Figure 20

SAMPLE TABLE SHOWING TRANSACTIONS USING DATA ELEMENTS

Data Elements	Admissions Form	Registration Form	Drop/Add Course Form	Grading Form	...	W 2 Form	Payroll Checks
For Students							
1. Name	X	X				X	X
2. I.D. Number	X	X	X	X		X	X
3. Sex	X						
4. Marital Status		X					
5. Residence		X					
6. Major	X	X					
7. Level		X					
8. Courses Taken		X	X	X			
9. Grades for Courses				X			
10. Grade Point Average							

component: that of input of additional descriptors. These must be added to the DED/DD system to describe the relationships expressed in the cross reference tables. Examples of such additional descriptors are: designator, synonym, user relationship, data creation (specification) relationship, transaction relationship, and potential user relationship.

7.3 Uses by Computer Software of the DED/DD System

The DED/DD system is also used by programming software. Some of these uses will now be discussed. This discussion could be skipped by an administrator not technically inclined.

7.3.1 Access Directory

In processing the computer sometimes uses cross reference tables generated from the DED/DD system. One example is a cross reference table which identifies the level of the security code needed by users for access to specific data elements and also specifies the type of access permitted, i.e. read or write access. In a terminal environment the data elements accessible by each terminal might also be specified in the table as well as the time of day when access is allowed. The table may also state limitations of the access time of each user and the user's account number for purposes of billing.

The cross reference table can be printed for the user's information in numerous sorting orders. One such ordering would identify the different data elements accessible to each person or security code. This arrangement is referred to as a Security/Access Directory. A sample of such a directory is shown in Figure 21.

Figure 21

ACCESS DIRECTORY

User Identification 076-835-5623

Access Limitation 13 hours (of CPU time for current academic year)
Account No. AS5842

Data Elements	Type of Access	Security Level	Terminal Number	Time Lock
Student Name	Read	10	04	08.00-17.00
Courses Taken	Read	10	04	08.00-17.00
Grades Received	Read/Write	12	04	08.00-17.00

7.3.2 Use in Processing

Before processing a data element it is necessary to have information on its form such as its mode of data representation, its format and its compaction (if it is compacted). Such information is usually provided in the relevant computer program but for data elements in a common data base it sometimes proves more economical to have this information in a dictionary that is machine readable. In such cases every time the data elements in the common base are used information necessary for processing is referenced from the DED/DD system.

A somewhat similar situation may arise in validating input data. If many values of the same data elements are entered from several points at different times, then a different validating program is required for each set of data. If the validating rules for each set of data were stored in machine readable form in a DED/DD then they could be referenced each time values of the data element entered the data base and the value tested for validity. If the data does not pass the validating tests then diagnostics of error are sent to the originator of the data. If, however, the data is valid then processing of the application programs continues. A graphic representation of this process is shown in Figure 22.

7.3.3 Locational Information

Once data is validated it must be stored and the stored location must be recorded to enable future access. Stored data constitutes a physical record. A physical record is distinct from a logical record which is a set of logically related data. Users are concerned with logical records (files). In order to access these logical records it is necessary to know the physical location of these records. This information is provided in a directory.

This conceptual problem is illustrated in Figure 23 (IBM, n.d.: 38). The relationships between the physical and logical records must be specified by the programmer in each application program or alternatively the mapping between logical and physical records can be done by the computer system provided the relationships between these records are defined in a directory.

In the directory method the programmer needs only to specify the logical records he wants. The physical locations of the records are determined by the computer system through the directory. The application program does not need to state the physical location of records used in processing. This is especially important in a multi-storage device system where the physical storage of data can be in one of many devices. The actual location is stated in the directory thereby relieving the applications program of having to specify the location of data.

7.3.4 Use to Generate Test Data

The rules for data validation discussed earlier include information on the character composition and the permissible range of data values. This information plus the information on format that is available in the data dictionary enables the generation of sample data needed for testing computer programs. Generating test data from the DED/DD enables the testing of programs even before live data is available. And even after live data is available the DED/DD generation of test data is useful for maintenance programs which can then be tested without disturbing the live data stream.

The sample test data generated by the DED/DD should be statistically designed (e.g. a stratified random sample) and, in addition, it should also include boundary conditions and exceptional values. All this information is available in the DED/DD system.

Figure 22

DED/DD DRIVEN CENTRALIZED VALIDATOR
FOR INPUT DATA TO DATA BASE

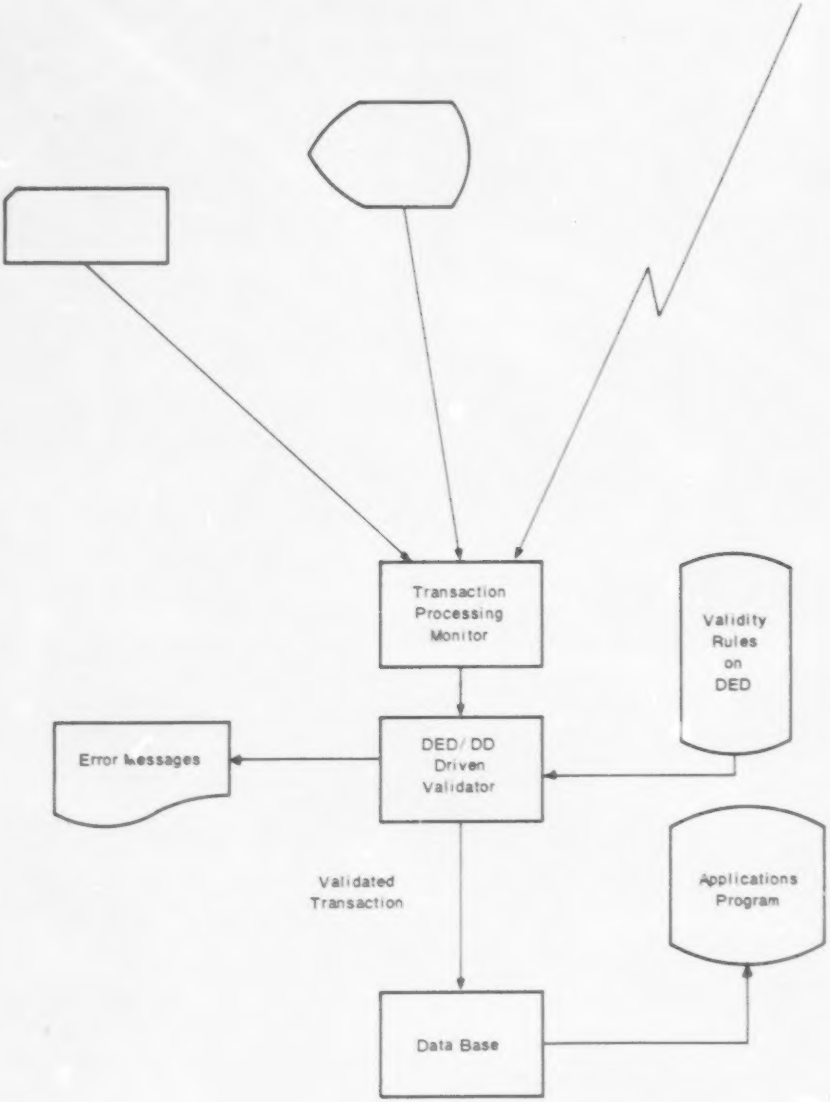
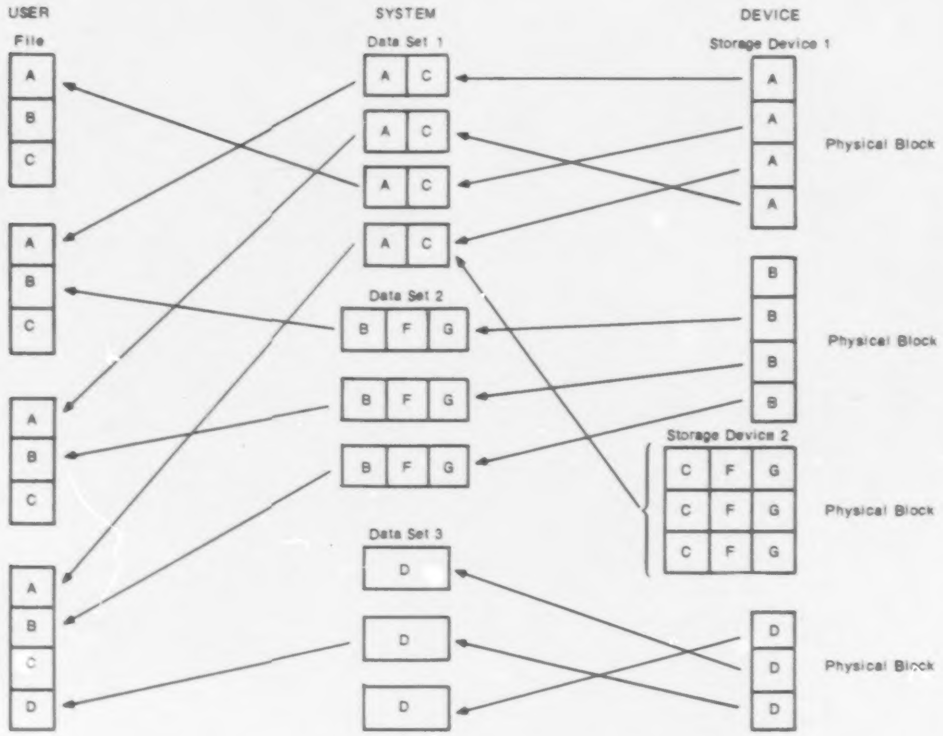

Transaction Processing Monitor
Validity Rules on DED
Error Messages
DED/DD Driven Validator
Validated Transaction
Applications Program
Data Base

Figure 23

LOGICAL TO PHYSICAL AND LOCATION MAPPING


USER
File
A
B
C
A
B
C
A
B
C
A
B
C
D
SYSTEM
Data Set 1
A C
A C
A C
A C
Data Set 2
B F G
B F G
B F G
Data Set 3
D
D
D
DEVICE
Storage Device 1
A
A
A
A
Physical Block
B
B
B
B
Physical Block
Storage Device 2
C F G
C F G
C F G
Physical Block
D
D
D
Physical Block

7.3.5 Code File

The dictionary part of the DED/DD system could be extended to include subclassifications of data elements. The names and codes of these subclassifications would constitute a separate file which would then be referenced every time data elements are used.

The contents of a code file is a subject of some debate. One school of thought argues that the code file should contain only codes that have many characters that are used very often. For example, the codes for names of courses offered are often long and these codes are frequently used in reports such as class lists, faculty loads, transcripts, course schedules, etc. Other examples of long and frequently used codes are the PCS codes and the Chart of Accounts. A code file would include these codes. The code file would not include codes like the sex code for though the code is frequently used it is short and therefore included in the computer program or defined externally.

In conflict with the above is the school of thought that argues that all codes should be in a code file to facilitate data definition control. It is argued that the needed extra equipment and the extra cost of processing required to include all codes in a code file is worth the control that can be exercised through such a file. This approach has been adopted in only a few institutions in the United States but abroad, as in Germany, all-inclusive code files are widely used.

There are some by-products of code files. They enable an updated listing to be generated by computer instead of pages of additions and deletions that must be updated in the typewriter or a MTST. These latter approaches are slow in response time and subject to error.

Another advantage of a code file is that it becomes a repository and authority on codes. Any new user wanting a code will refer to such an authority rather than create his own codes. This reduces the possibility of duplications and synonyms that would result in confusion and inefficiencies.

7.3.6 Descriptors Needed for 7.3.1-7.3.5

The uses discussed above require additional descriptors. These are: codes (for subclassifications of data elements), mode (bit string, character string, packed decimal, etc.), logical/physical relationship, security code (for reading and writing), and medium (of storage)(1).

---

1) For another list and one that is very formally described, see Plourde (1973: 707-712).

The above is the third list of descriptors discussed in this paper. (For the other two lists, see 5.2 and 7.2.5). There could be additional descriptors. One organization is known to have 250 descriptors. They cover all the needs of COBOL, Assembler source programs, JCL, operating system and software packages being used (Canning, 1972: 4). This, however, is an exceptional case.

8. SUMMARY AND CONCLUSIONS

Data is a resource to institutions just as buildings and employees are organizational resources. Data must be organized, managed and controlled. This is done by a Data Element Dictionary/Directory System. This system describes, protects and facilitates the use of a common data base which is a set of integrated data used by more than one current or potential user. The integrated pooling and centralising of the data improves the efficiency, flexibility and control of the data so that it can be interpreted consistently and made available to all authorised users.

The first basic component of the DED/DD system is the DED. It identifies all data elements in the common data base. Some data elements to be included in the common data base are determined by the Input-Output Matrix. Other data elements in the common data base are required and specified by the analyst such as data elements for identification, linking, reference, checking totals, security control, and planning. Additional data elements are required for classification, the most important of these being the Program Classification Structure.

The DED is the authority and formal repository of information on data elements. Such information is done by descriptors. A list of frequently used descriptors is shown in Figure 24.

The descriptors are needed for processing. They also provide a basis for establishing and reinforcing necessary standards and responsibilities relating to data collection, processing and usage.

A DED also contains descriptions of the data sets, logical records, and file definitions. When changes in these sets, records or definitions need to be made, one change in the DED is all that is necessary for all applications programs that would be affected by these changes. The DED also contains information on the physical location of data. This relieves the user of the problem of keeping track of the physical location and storage device of data. This information on location is stored in tables, sometimes referred to as directories, which are used during processing. Other indexes, tables and directories contain information which relates data elements to their origin, transactions, key words, users, uses and access information. Most of these relationships are summarised graphically in Figure 25.

Figure 24

DESCRIPTORS NEEDED FOR A DED/DD SYSTEM

Name (for ease in identifying and remembering)
Definition
Derivation algorithm (including accuracy level)
Units (where relevant, e.g. dollars, FTE)
Format
Justification
Width of field
Validity rules

Identification
Version of issuance
Date of issuance
Status (effective, proposed or approved)
Last date changed

Designator
Synonym
User relationship
Creation/specification responsibility
Transaction relationship
Potential use relationship

Codes
Mode
Security code
Billing code
Access limitation
Logical/physical relationship
Medium of storage

There are by-products of the DED. For example, the Test Data Generator which simulates data can be used for testing applications programs (new and old) without using live data. Also, a Table Driven Validator which validates input data uses validity rules defined in the DED.

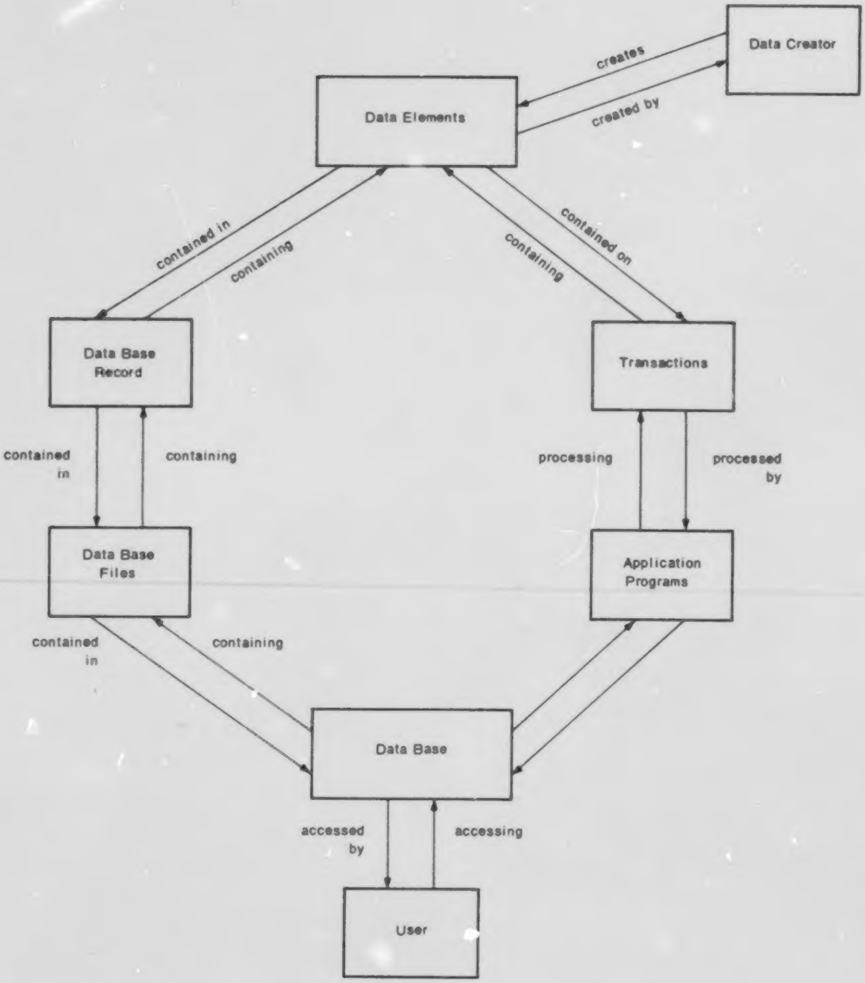
The uses and by-products discussed above and their respective users are shown in Figure 26.

The DED/DD is important for the development, use and maintenance of applications programs in a computerised data system. But the DED/DD is the repository of information not only for computerised data but also for some data that is used in manual operations. Therefore the problems of organizing a DED/DD become difficult.

A data base is only one component of an information system - albeit an important one. Other components are: forms and procedures for collecting the data and using the output: equipment for processing the data; computer programs that instruct the computer on how to process the data; and finally, documentation on the information system itself so that it may be properly operated and maintained. These components of an information system and their interfaces for their integration must be so designed that they generate the desired information. Approaches to such a design is the topic of the next chapter.

Figure 25

SOME RELATIONSHIPS DOCUMENTED BY THE DED/DD


Data Creator
creates
created by
Data Elements
contained in
containing
containing
contained on
Data Base Record
Transactions
contained in
containing
processing
processed by
Data Base Files
Application Programs
contained in
containing
Data Base
accessed by
accessing
User

Source : Adapted from Curtice (1974 : 102).

97

Figure 26

USERS OF DED/DD SYSTEM

	Non-data processing user of Data	Data processing management	Systems analyst	Programmer	Coder, data preparer	Data administrator	Computer/program
Generalised documentation	X		X		X	X	
Validity/edit rules	X		X	X		X	X
KWIC/KWOC indexes	X	X	X	X	X	X	X
Uses/users directory	X	X	X			X	
Data creator directory	X	X	X		X	X	
Transactions directory	X		X			X	
Access directory		X				X	X
Processing information						X	X
Relational information						X	X
Code file						X	X

REFERENCES

ANSI (1970), American National Standard Vocabulary for Information Processing, X3.12-1970 (N.Y.: American National Standards Institute).

Cahill, J.J. (1970), "A Dictionary/Directory Method for Building a Common MIS Data Base", *Journal of Systems Management* (November), pp. 22-29.

Curtice, R.M. (1974), "Some Tools for Data Base Development", *Datamation*, Vol.20, No.7, pp. 102-106.

Daniels, A. and D. Yeates, (1971), Systems Analysis (Palo Alto, California: Systems Research Associates).

Canning, R.G. (1972), "The Data Administrator's Function", *EDP Analyser*, Vol.10, No.11, pp. 1-14.

Canning, R.G. (1974a), "The Current Status of Data Management", *EDP Analyser*, Vol.12, No.2, pp. 1-14

Canning, R.G. (1974b), "The Data Dictionary/Directory Function", *EDP Analyser*, Vol.12, No.11.

Goddard et. al.(1973), Data Element Dictionary, 2nd Edition (Boulder, Colorado: NCHEMS at WICHE).

Gulko, W.W. (1972), Program Classification Structure, 1st Edition, Technical Report No.27 (Boulder, Colorado: NCHEMS at WICHE).

Hussain, K.M. (1973), Development of Information Systems for Education (Englewood Cliffs, N.J.: Prentice-Hall, Inc.).

Hussain, K.M. and J.S. Martin (1971), A Resource Requirements Prediction Model RRPM-1: Report on the Pilot Studies (Boulder, Colorado: NCHEMS at WICHE).

IBM (n.d.), Introduction to Data Management Student Text, SC 20-8095-0.

Lyon, J.D. (1971), "The Role of the Data Base Administrator", *Data Base*.

Mason, R.R. (1973), "The Purpose of Analytical Models: The Perspective of a Model User", Paper presented at the CSCUP Conference.

NCHEMS (1974), Inventory of Statewide Measures (Boulder, Colorado: NCHEMS at WICHE).

Parker, D.B. et al. (1973), Computer Abuse (Menlo Park, California: Stanford Research Institute).

Plourde, P.J. (1973), "Elements of Data Dictionary Systems Design", in *Proceedings of the 1973 National Meeting*, pp. 707-712.

Thomas, C.R. (1970), Data Element Dictionary: Student-First Edition, Technical Report (Boulder, Colorado: WICHE).

Topping, J.R. (1972), Cost Analysis Manual, Prelim. Ed. (Boulder, Colorado: NCHEMS at WICHE).

Topping, J.R. and A.K. Miyataki, (1972), Program Measures (Boulder, Colorado: NCHEMS at WICHE).

Traver, C.M. (1973), "Data Base Administrator - The Emerging Position", Proceedings of the College and University Machines Records Conference.

Uhrowczik, P.O. (1973), "Data Dictionary/Directories", IBM Systems Journal, Vol.3, No.4.

Willoughby, T.C. (1972), "Staffing the MIS Function", ACM Computing Surveys, Vol.4, No.4, pp. 241-259.

Chapter 5

DESIGN OF AN INFORMATION SYSTEM

by

James Farmer
Systems Research Inc.
Oklahoma, U.S.A.

CONTENTS

1. Approaches to System Design 104
2. Data Processing Design - Approaches to Application Programming 105
3. Design Approaches to Communications 110
4. Design Approaches to Data Organization 112
5. Criteria for a Data Processing Design 114
6. Flexibility 116
7. Characteristics of a Good Systems Designer 118

1. APPROACHES TO SYSTEM DESIGN

If one views the administrative functions, or major administration functions, as a whole system, there are several alternative approaches to designing data processing systems to support these functions. Some institutions view their administrative functions as single systems and are attempting to develop all of the possible interrelationships of such systems. Others are taking major administrative functions such as personnel and payroll, student records, registration, scheduling, and student financial aid, or accounting and costing, as a single system.

As shown in Figure 1, there are basically three approaches which may be taken to designing such a system. The evolutionary approach suggests a schedule of implementation for the various components of a system and designing, developing, and implementing each component, or subsystem, one at a time. For example, a student financial aid system could consist of a student application subsystem, a student financial aid accounting subsystem, a student financial aid packaging subsystem, and a student financial aid special reports subsystem. With the evolutionary approach, the first subsystem, which would likely be called a "system", would be the student financial aid application system. Its design would be independent of the design of any future systems except that the knowledge of the designer would likely include some provisions for the future system. Later subsequent systems would be designed, developed and installed. This approach has one significant advantage over all other approaches - early results. Since the design of the specific subsystem, or system, does not have to bear the time delay or cost of designing a complete system, an early operational capability is achieved.

Figure 1

SYSTEM DESIGN APPROACHES

- Evolutionary
- Modular
- Single System

A second approach is to design all of the subsystems into a system with a single design, but plan on developing and implementing the subsystems sequentially. This system design approach differs from the evolutionary approach primarily by having a single design. This approach reduces the modifications, or accommodations, which must be made between the subsystems when they are later integrated into a single system. The design focuses on the interactions to assure that data is appropriately available to each of the subsystems or "modules".

A third approach is to design, develop, and implement all of the system components as a single system. The primary advantage of this is a reduction in the transition time - there is only one period of change, and purportedly economies of scale and integration. In practice, the single system or "big system" approach has not succeeded well. It tends to be expensive and requires considerable time from the initial design to the initial operational capability. There are examples of military systems, government systems, and some commercial systems - like airlines reservations systems - which are designed and implemented under the "big system" approach. As evidence now shows, there are diseconomies of scale in computer programming, and impatience on the part of users to obtain the benefit of their investment at the earliest possible date.

The consensus in education circles appears to be on modular system designs. It should be noted, however, that rapid change both in data processing technology and educational institutions will cause design changes as development and implementation occur, even with the modular approach.

2. DATA PROCESSING DESIGN APPROACHES TO APPLICATION PROGRAMMING

It is useful to attempt to classify the design approaches for data processing systems, or "application systems". The term "application systems" is often used in data processing to differentiate between applications and the operating systems, but here the term "system" refers to an application system - a collection of procedures, computer programs, and instructions for operation - rather than an operating system or hardware.

As shown in Figure 2, there are three basic approaches to designing data input to a system. These include an interface with some machine readable files. Many of the current applications now include such interfaces. Student information is provided by files from the college testing services, student financial aids from the college admissions or student records, and so on. For planning and management many of these files are provided by agencies external to the college and university such as those data on social and demographic characteristics or surveys of potential students.

Most of the data input is, however, "transactions". These usually are data prepared from some input document such as student's applica-

Figure 2

DATA PROCESSING DESIGN APPROACHES

APPLICATION

- Input
  - Interface with Files
  - Transactional Processing
  - Conversational
- Output
  - Query Language
  - Report Writer
  - Custom Reports
- Application Packages

COMMUNICATIONS

- None - Batch
- Remote Job Entry
- Interactive
  - Conversational
  - Transaction Processor

DATA ORGANIZATION

- Files
- Data Base Access Systems
- Data Base Management System

tion, an instructor's grade report form, or a controller's receipt. Often there is an intermediate step of converting these "transactions" to machine readable form. This step may be keypunching cards, or key-to-disk or diskette data entry. The method of entering transactions "on-line" is becoming more popular. Usually a terminal with a keyboard and video-screen is used for this purpose. The primary advantage of such system of data entry is not cost, but rather the ability to check certain aspects of the transaction with available information. For example, the account number can be checked with the account file to determine if such an account actually exists. A name can be checked to be sure that it contains no numeric data and conversely an amount field can be checked to see that it is a valid numeric quantity. The results of these edit checks are immediately made known to the input operator so that a correction can be made while the input form is in front of the operator. Also the results of a complete transaction, such as a name and address, can be displyed to the operator for checking. The rate of data input for such systems is typically 5 to 10 per cent slower than for data conversion. Part of this comes from the response time of the terminal - it may take 0.5 to 2 seconds to respond to input - and another part comes from the checking or correction of the operator. However, the error rates are reduced by as much as 90 per cent and there is no need for computer runs to perform such editing. This may result in a savings of 10 to 25 per cent of a typical data processing application in colleges and universities.

Transactional processing can also include the use of machine-readable source documents. These include checks, receipts, and other financial transactions with the magnetically coded special characters, documents with special marks and characters for optical recognition, and pre-punched cards. More recent input includes universal product bar codes and the coloured bars of the transportation vehicle codes used for trucks and trains.

A third approach to data input is a "conversation" between the operator and the computer program. This method of data entry is often used when the person providing the information is operating the terminal. The questions are a combination of the questions of a "form" and the edit queries of the edit routines. This method is being extensively used now to develop medical histories, applications for government assistance, and computer aided testing and instruction. Such an approach may also be used by untrained operators who are collecting data from interviewees or on the telephone.

The most user-oriented of the methods of producing data from a data processing system is a query language. A query language is typically used in the conversational mode to obtain data from a file. Figure 3 shows a query and the results of a typical information request. The query, which would be typed on a keyboard terminal, is "PRINT FICE INSTNAME ENTOT WHEN STATE = CA AND CARNCODE = 11". This query selects those institutions on the file which have a Carnegie Commission classification of 11 - major research universities, and a state code of CA - for the state of California. The file is searched for records which meet these two characteristics and then prints, on a single line, the US Federal Interagency Code for the institution, the institution name, and the total enrollment. A table provides for each file the record location of the elements and the column headings. The table also gives the type of data - alphabetic, numeric or special character - and whether a total or average is appropriate for numeric data. If the query language is operating in conjunction with a data base management system (DBMS), the table is incorporated into the DBMS or data pointers are used in lieu of locations. The query language, which has been often developed as part of the application rather than as a separate general language, is intended to meet most needs for data without the need for a computer professional to operate the system. This is a particularly important capability to have if an organization has only machine-readable files.

Figure 3

QUERY EXAMPLE

"PRINT FICE INSTNAME ENTOT WHEN STATE = CA AND CARNCODE = 11"

```
INST             INSTITUTION                FALL 71
FICE             NAME                       TOTAL
CODE                                        ENROLLMENT

1131    CAL INST OF TECHNOLOGY               1,504
1305    STANFORD UNIVERSITY                 12,479
1312    U OF CAL BERKELEY                   28,384
1313    U OF CAL DAVIS                      14,571
                  .                             .
                  .                             .
                  .                             .
                                           180,400
```

A report-writer is a computer program which interprets descriptive statements into computer instructions for preparing a report. Report-writer commands are similar to the steps used by a programmer in developing a program for a report. For example, one would specifically identify the text for the column headings and the class of the data and its field width-information which is provided by the table for a query language. On the other hand, the report writer has a number of capabilities - like totals for different levels, page format, code conversions - which are not available to current query languages. It is likely that report-writer features will be included in query languages to form "conversational" report-writers as a future programming method.

It is always possible, and sometimes required, to use a programming language such as COBOL, FORTRAN or ALGOL to prepare reports. The format of these reports and their contents are almost unlimited, but requires more effort than if a report-writer or query language could be used. (However, some programmers are sufficiently proficient that they can use a programming language as fast or faster than a report-writer, and have no limitations on format).

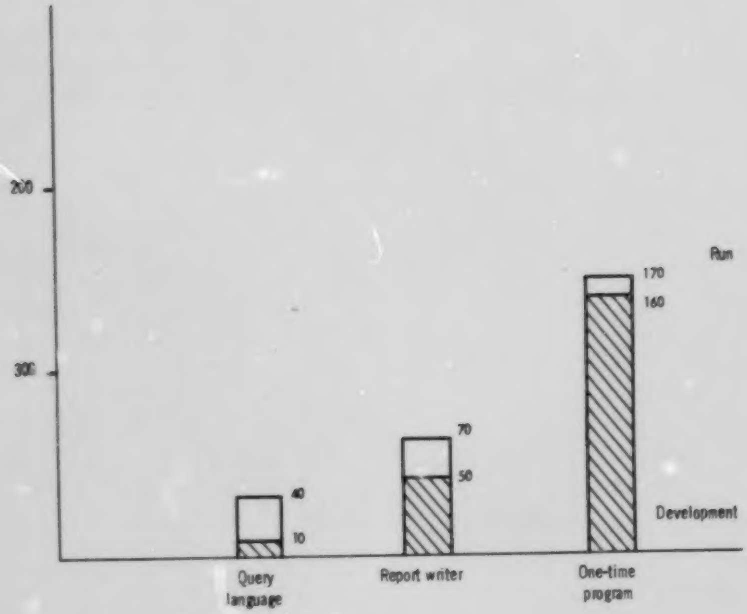
Figure 4 shows the typical cost of preparing a set of reports using the three alternative approaches. These average costs are for a set of 4 to 15 reports on six to twelve variables using special HEGIS files(1) of United States Federal Government data on about

---

1) Higher Education General Information Survey of the National Center for Education Statistics.

Figure 4

COST OF REPORTS USING ALTERNATIVE PROGRAMMING STRATEGIES
(small files)



2,800 United States colleges and universities. These special files have one record per institution with six to seven hundred data elements. Clearly there are a number of variables which affect the comparative prices of these approaches to report preparation, but these results show the savings that can be achieved if a data file management or data base management system has been implemented and a query language is available.

Sometimes a design is developed including an existing application package. The design usually modifies the package input to correspond to existing or proposed forms, and prepares computer programs for special or additional reports. Because of the number of application packages available in higher level languages and the economics, this approach is becoming more frequently used.

Typical designs using application packages include class scheduling, accounting, student financial aid loan processing, grade reporting, and payroll. Often interface programs are used for data input to these systems to preclude package modification. The output files can be used with query languages or report-writers to achieve further generality and additional report programs can be used to achieve specific report formats.

3. DESIGN APPROACHES TO COMMUNICATIONS

Because of the decreasing unit cost of communications and the increasing availability of telecommunications software and equipment, the cost-effectiveness of on-line applications is increasing. The option of using a communications based design is available for almost all systems.

For large volume of print-out or input transactions - such as checks to a bank - the traditional batch processing design, without using communications, is both effective and low cost. As sharp price differentials for computer time are being implemented for night and week-end work and as micro-fiche applications are being used to avoid the high cost and inconvenience of large reports, the batch processing approach is viable for a large number of applications.

Because remote job entry - terminals with card readers, printers, and sometimes tape drives - costs only slightly more than batch processing, this approach is particularly useful to organizations which have components located geographically apart. In fact, only a few miles or few buildings' separation often creates sufficient travel time and inconvenience to suggest remote job entry. Since some of the operational tasks for a remote job entry terminal are performed by the user rather than the operational staff of the computer centre, it may be that remote job entry is less expensive than local batch operations. As a user organization matures in its use of data processing usually the skills necessary to initiate and control data processing applications are found in the user organization. When this occurs, then remote job entry is an appropriate alternative for a system design. If the system design is to support remote job entry, then the processing can be designed to have more interaction with the user. For example, if accounting is developed as a remote job entry application, the edit runs would be separated from general ledger and journal reports to assure that the user could review control totals before proceeding. Such control totals may have little meaning to a data processing operator. Similarly some reports which are needed periodically can be separated on on-demand processing by the user whenever the report is needed. While there are no major differences in the design between a batch and a remote job entry terminal application system, there can be much more user interaction with the system.

Communications with keyboard terminals are becoming increasingly effective. Terminals which print 30 to 60 characters per second (300 to 600 words per minute) and have full character sets and full-width (132 print positions) carriages, are both common and inexpensive (less than US $100 per month). Terminals with video screens having 24 lines of 80 characters are also common and inexpensive (less than US $80 per month). With prices approaching the cost of typewriters, interactive processing becomes more attractive.

For purposes of design, there are two types of interactive applications. One, the conversational application, is similar to the interactive compilers such as BASIC and APL. They respond to the user at the end of each line, or series of lines. This approach is used for query languages and conversational input or questionnaire input. The system can imitate a transaction processor by requesting data and providing edit messages similar to those of the typical edit program.

A transaction processor is a computer application usually designed using a video screen and a special computer operating system which minimizes response time. This is typically used for a large volume of transaction input and provides limited symbols or mnemonics to identify the needed data or errors. These systems are usually used where a single data entry operator would deal with a limited number of types of transactions and would, therefore, become particularly proficient at that specific application. These systems are particularly effective for large volumes of transactions in government - for licences, tax returns and taxes, and welfare and social programs, and in large businesses - insurance policies, medical claims, and magazine subscriptions. Because immediate response time is required, since costs are directly proportional to response time, these applications are usually done at a location near the processing computer and the terminals are connected to the computer over high speed communications lines. There are improved hardware and software and communications techniques which are now permitting these kinds of applications requiring high speed communications to be done at distant sites without requiring a high speed communications line for each terminal. This permits data entry applications with four or more terminals to operate almost as inexpensively as if they were located near the processing computer, but the design does not lend itself to geographically distributed data entry.

4. DESIGN APPROACHES TO DATA ORGANIZATION

Traditionally data processing has organized data into files of records. These files may be processed sequentially - one record following another, through an index which gives the approximate location of a record - such as the indexed sequential access method, or directly where a record's physical location can be determined from the value of a data element called the key. The computer subroutine for processing requests for data are called input-output subroutines if they are incorporated into the program or data access methods if they are incorporated into the computer's operating system. Such routines are included in most programming languages such as COBOL, FORTRAN, or ALGOL. Using such data requires that the computer programmer know the physical location of the data in a record, and in which file the record is found.

The physical location of data can be given in a table with the name of the data. Thus, a user or an application programmer can refer to data by name and the application program, including a query language or report-writer. The relationship must be defined only once - when the table is prepared. Such systems could be called data base access systems.

It is also possible for a computer program to prepare the table and select a physical location appropriate to the characteristics of the data and relationship between data elements. These systems are often called data base management systems. Such systems as IMB's IMS, Univac DMS and System 2000 are data base management systems.

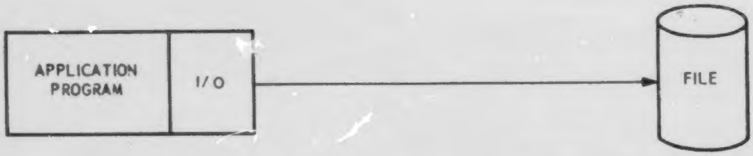
Conceptually the four methods of data access are illustrated in Figure 5. The application program and "definer" tables are prepared by the application programmer while the data access routine (or data access methods) and data base management systems are part of the computer's operating system.

Input-output routines are the most efficient in the use of computer time, but require detailed and tedious programming. On the other hand, data base management systems require a large number of computer instructions - often several thousand - in order to locate

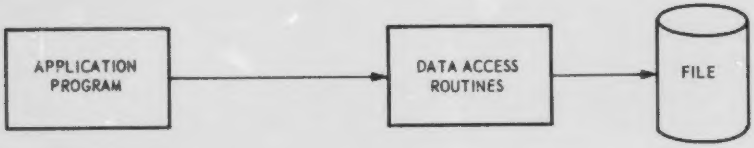
Figure 5

CONCEPTUAL METHODS OF ACCESSING DATA

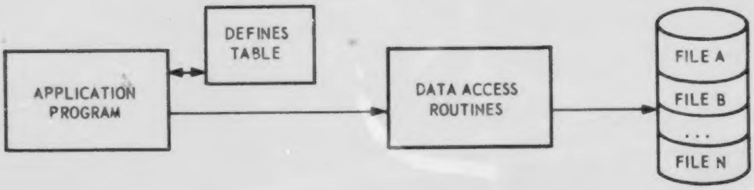
INPUT-OUTPUT ROUTINES


APPLICATION PROGRAM
I/O
FILE

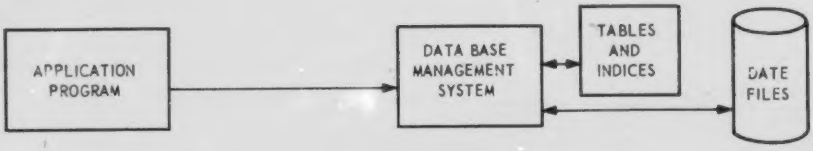
DATA ACCESS ROUTINES


APPLICATION PROGRAM
DATA ACCESS ROUTINES
FILE

DATA BASE ACCESS SYSTEMS


DEFINES TABLE
APPLICATION PROGRAM
DATA ACCESS ROUTINES
FILE A
FILE B
...
FILE N

DATA BASE MANAGEMENT SYSTEM


APPLICATION PROGRAM
DATA BASE MANAGEMENT SYSTEM
TABLES AND INDICES
DATE FILES

a data element, but do not require so much programming time or design effort as the detailed application program. The choice of the appropriate method for assessing machine readable data is a design decision.

Figure 6 shows some examples of software which support specific data access approaches and communication methods. Some of these components include manufacturers' supplied software, proprietary systems and applications programs. The various combinations have markedly different performances, cost characteristics and history of reliability. All such combinations should be considered when designing a system.

5. CRITERIA FOR A DATA PROCESSING DESIGN

As shown, there are a number of different approaches to design and most could be applied to a specific application and the resulting system or subsystem could be operational. The problem is to select an approach, and further a design, which is preferred to others. Six criteria for selection are shown in Figure 7.

The most important criteria is functional suitability - does the application system design perform the specific functions needed by the user? The user can best judge this characteristic of a specific design provided he understands explicitly the full range of alternatives. The computer specialist is responsible for providing the user with sufficient information to make this judgement.

An evaluation of the value of timeliness must also be made by the user. This refers both to the initial operational capability of the system and to the processing schedule. In order to make an approximate evaluation, the user must have available the full range of alternatives and costs, but the designer must have the constraints on timeliness. Most application systems have externally imposed schedule commitments. For example, the school term affects student registration and faculty appointments, and payday affects payroll.

Usually data processing can significantly improve timeliness; however, frequently the applications are judged by their users to reduce timeliness. This primarily occurs when the detail of transactions becomes "lost" in the system at the same time as summary reports are made available more quickly. The user should be careful, in guiding the design, to evaluate the effect of timeliness on all functions of an organizational unit.

Many of the services which are provided to students or faculty and staff include administrations as a component. For example, a student must participate in registration as part of his course work. The student's attitude toward the institution may be severely impacted by administration. The improvement in its administration may change the way a student views the university. With American community

Figure 6

EXAMPLES OF SOFTWARE COMBINATIONS

		Communications		
		None (Batch)	Interactive Conversation	Interactive Transaction Processor
Data Accesses Approaches	Data Access Routines	Cobol Fortran RPG	Basic Interactive Fortran	CICS
	Data Base Access	Mark IV Cobol and Total	DS/3 TSO and DLA	
	Data Base Management System	Cobol and DL/I	IMS/IQF	CICS with DL/I

Figure 7

CRITERIA OF A GOOD DESIGN

- . Functional Suitability
- . Timeliness
- . Process or Product Enhancement
- . Reliable Performance
- . Cost Benefit
- . Flexibility for Growth

colleges experiencing enrollment increases of 5 to 10 per cent because of on-line enrollment systems, which make registration much less burdensome, it is clear the potential effect the administration systems can have on a university's program. This could be viewed as process or "product" improvement.

A system must also produce reliable performances when installed. Reliability is not just a function of the computer program. Training of operational staff, testing, and management understanding all contribute to reliability. During the design it is necessary to choose approaches which enhance reliability. This includes methods of error checking and cross-checking data; relations of the system to user activities - the man-system interface, and design of information feedback.

Cost benefit is an obvious, but perhaps not the most important criterion for evaluating a system design. Cost benefit should include clerical savings, if any, potential improvement of management decision, and any benefit from previous "product" improvement. The data should include all costs for design, development, installation, testing, conversion and implementation, and operation. The costs should include both incidental and total costs. Since some of the benefits are judgements, they should be imputed identifying the underlying assumptions or source of the estimates.

One system design criterion which is difficult to include is flexibility. Because colleges and universities are constantly changing, the applications must be capable of change. Usually a system which is designed for frequent change is comparatively inefficient. However, rather than attempt to stop change - where the decisions of a college and university are dictated by data processing systems limitations, or constantly replace efficient software, it is often preferable to use inefficient approaches which accommodate change gradually as it occurs. Since computer time, compared to application development, is inexpensive, most colleges and universities are placing emphasis on flexibility.

These criteria cannot be well quantified. Evolution depends upon subjective judgements. It is important, however, that the evolutions include at least these criteria and a special effort should be made to keep the evolution of a system design broad.

It should be clear that the resulting system design selected will differ if all of these criteria are used in lieu of a narrow criterion. The resulting systems tend to be more flexible, to support management, and to place more emphasis on person-system interface. Each of these tends to increase the use of computer time, the use of generalised software, and the use of user keyboard terminal systems.

6. FLEXIBILITY

Because of the emphasis on flexibility it may be useful to identify the system design techniques which increase the flexibility of an application system. Some are listed in Figure 8.

One approach is to include potential future functions in the software design. The functions are then implemented as needed. When future changes or added functions are known, this is a viable approach to improved flexibility. When training is the primary constraint, and the software development investment is possible, inclusion is a viable approach.

Another but similar approach is to provide the application system with the interface, often called "hooks" which permit functions to be added to the system without having to redesign or modify the basic application system. If a designer has good guidance on the likely nature of future changes - e.g. new operating requirements, change in university organization program, data requirements for a model - he will be able to include provisions for future expansion. Usually this means that the design will include blocks made in records for new data elements, more steps in the process, and possibly a different file structure.

Another technique is to deliberately design the system for computer size and resources less than those actually available. This permits later expansion without redesigning. For example, a system may be designed for 20,000 student records when there are only 15,000; a record length of 200 when there are only 120 characters of data; or program length of 50k bytes when there are 64k bytes available; or using two tone files when there are more available. Had all resources been used, it would have been necessary to modify the system and its program to add or change a function. Again, the inefficiency of using less than all of the available resources reduces the cost of changing the system in the future.

Figure 8

APPROACHES TO FLEXIBILITY*

. Inclusion of Functions
. Anticipation
. Expandibility
. Modular Design
. Parameterization

---

* Adapted from Frost, David "Designing for Generality", Datamation Vol.20, No.12, December 1974, 70, 59-61.

Perhaps the most popular approach to flexibility is modular design. In this case each data processing function is placed in a separate program or sub-program. As the functions change, the modules are changed or added. For example, in a modular design one program may be used to calculate payroll while another is used to print the checks. Then a change in payroll computation can be made without changing the report program, and similarly the format of check printing program or report program can be changed without changing the calculation program.

Parameterization is a method of changing the processing by changing some variables supplied to a program. For example, the tuition or board and room charges can be supplied on an external table rather than being contained in the program as specific value. Therefore, if tuition or board and room charges were changed, the table would be changed, not the program. Parameterization, as a technique, is powerful and is being used, to some extent, in all current system designs. It presents an additional operational function - maintenance of the table - which is transferred from the programmer or program maintenance, to a user as a data file maintenance task.

7. CHARACTERISTICS OF A GOOD SYSTEM DESIGNER

A college or university administrator depends upon the system designer. Often it is difficult for a manger to judge the quality of the resulting design or competency of the design, or a manager would like to know the characteristics of a good system designer. It is not possible to provide a guide for evaluating system designers, but it is possible to hit some of the characteristics which are found in most system designers who are considered good.

Increasingly the most important characteristics of an application system designer is knowledge of the application area. As the functions represented by the data system become more complex and the interaction between the user and system analyst more direct, knowledge of the application area is important to a design which includes the most recent advances in data processing applications. Knowledge of the application area is also important to communicating with the user and encouraging his participation in design activities and his thorough criticism of the design.

Since the system designer is the primary source of design alternative, a good system designer has a large repertoire of techniques on which he may draw. In this activity, the creativeness of the designer is most important. No particular background - theoretical preparation, experience, or user - seems to consistently produce the best designers. Perhaps creative designers are, like artists, born with the talent.

Data processing skills are, of course, necessary. But these skills are not associated with a specific programming language or knowledge of a specific operating system, but rather on in-depth knowledge of the concepts and principles and sufficient knowledge of the specific language and operating system to recognize constraints and make design choices appropriate to the specific hardware and software.

Because of the necessity for a designer to formulate and make trade-off decisions, a designer must have an intuitive knowledge of economics. The designer must implicitly and explicitly understand and practise "good economics" by making consistently rational decisions.

As many experienced data processors have observed, the needs of users, particularly managers and executives, will change as the users mature in their use of the results of a data processing system. If a designer is aware of this process he will design systems which accommodate this change. Hopefully, a system can, without modification or addition, produce the kind of information which will be needed.

It is not possible to abstractly specify the characteristics of a good designer, but administrators and managers should be aware that a number of skills beyond mere computer hardware and software competency, which contribute to the system design process, are necessary.

Figure 9

CHARACTERISTICS OF A GOOD DESIGNER

- Knowledge of Application
- Creative Alternatives
- Data Processing Skills
- Economic Intuition
- Familiarity with User Maturation

After a system is designed, it must then be implemented in order to make the design operational and workable. This involves the selection and acquisition of necessary equipment (if any is desired); the ordering and delivery of forms required for input and output generation; and the writing of the computer programs necessary for generating this desired output. These topics are well documented in the literature on information systems and will not be discussed further. Instead we will discuss the next stage in the development of an information system: the testing of the system and the conversion of the old system to the new one. This is the topic of the next chapter.

Chapter 6

TESTING AND CONVERSION OF AN INFORMATION SYSTEM

by

K.M. Hussain

CONTENTS

1. Introduction 122
2. Planning of Test 122
    2.1 Specification of additional load 123
    2.2 Reliability specifications 123
    2.3 Performance standards 124
3. Testing Strategies 124
    3.1 Pilot testing 124
    3.2 Parallel run 125
    3.3 Simulation 126
4. Testing Considerations 126
5. Test Data Preparation 126
6. Levels of Testing 127
    6.1 Component testing 127
    6.2 Functional testing 128
    6.3 Subsystem testing 128
    6.4 System testing 128
7. Testing Responsibility 130
8. Test Schedule 131
9. Conversion and Related Activities 132
10. Summary and Conclusions 136

References 140

Selected Annotated Bibliography 317

1. INTRODUCTION

The testing process involves the comparison of desired performance (as stated in the User's Requirement Specifications) with actual performance, the identification of deviations (if any) and its elimination. Once the system performs according to specifications, then the system conversion takes place which is the phasing-out of the old system and the phasing-in of the new system. These and related activities are the subject of this chapter.

The order of the topics to be discussed is as follows:

Planning of Test
Testing Strategies
Testing Considerations
Conversion and Related Activities

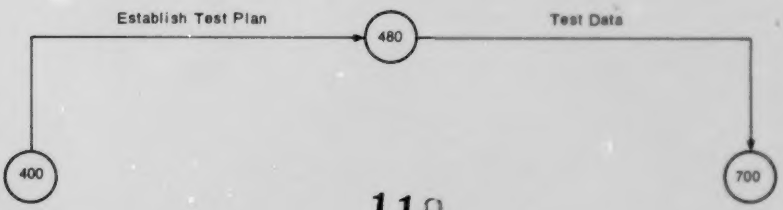
2. PLANNING OF TEST

The planning of the systems test is the first of the testing activities and is shown on the partial network in Figure 1 as activity 400-480. This is part of the set of activities performed in parallel to the programming and physical planning (not shown in Figure 1) but discussed later in preparation for the testing stage. Activity 480-700 will also be discussed later.

Planning for testing involves specifying the test specifications as well as selecting testing strategies. First to be discussed are the testing specifications.

Figure 1

PARTIAL NETWORK SHOWING SOME TESTING ACTIVITIES



Test specifications include: specification of additional load as well as reliability and performance standards. Each will be discussed below:

2.1 Specification of Additional Load

If a factor of safety is not specified or if it is specified but does not allow for expansion, then such an allowance must be made. Growth in load for the lifetime of the system must be estimated by statistical methods of estimation and the system must be tested for such estimates.

Besides expected expansion, other additional loads could be caused due to peak periods. For example, testing the data preparation and processing capabilities for the response time of a job during normal loads is insufficient if the job is required during peak periods. In educational institutions, peak periods come every semester. An example is the work of the Registrar's Office. At the beginning of the semester they must compete for services with other offices involved in registration. If registration falls at the end of the month, there is the competition of processing with the Business Office doing their end-of-month jobs. Thus, peak-period loads are important.

Another related situation of load testing may occur in cases of emergency. What happens if the equipment breaks down for a few days during registration or at the end of the month when the pay checks are produced? The administrator may decide that he cannot afford such a breakdown and that the system must be operated in spite of such breakdown. A solution to this problem has been found by making the system operational on an accessible system (sister educational institutions are not always helpful for registration breakdown because their peak periods coincide!), which requires additional testing on the "standby" system. Another solution to this problem is to have some slack in the schedule to allow for a short breakdown. Whichever solution is adopted, it should be tested to ensure that it can meet the required performance levels.

2.2 Reliability Specifications

Reliability specifications refer to allowable occurrences of errors in processing. An example would be the allowance of one grade error in the processing of a thousand grades. This specification is of great interest to the administrator and this judgement must be made with great care for it involves important trade-offs of effort and equipment.

For example, consider grade processing again. The testing for reliability may find this to be less than the reliability specifications set. The system may then have to be redesigned or alternatively, the testing conditions may have to be re-established. This is an example of the iterative recycling nature of the development process: design, test, redesign, retest and recycle till the test is satisfactory or: systems specifications, design, test, restate systems specifications, redesign and retest. (Some steps in the development process have been excluded to emphasize the relationship between design and testing.)

2.3 Performance Standards

An example of performance standards may be the specification of response-time. In the grade processing example, the test specification may be the processing of grade reports to be mailed to students within 48 hours of the last exam.

As with the reliability specifications, performance specifications also have important trade-offs of labour and equipment and may require recycling of the development process.

The specifications of performance reliability as well as the work-load for specific system components or subsystems, may be specified as one document in which case it must be done after all components and subsystems are designed, or it can be done in general terms (covering such factors as growth rate and emergency conditions) in addition to the test specifications being made with each component and subsystem design. The choice is largely a function of the complexity of the system and the background and preference of the system design team.

3. TESTING STRATEGIES

Once standards are specified, a strategy of testing must be selected. There are three main strategies of testing:

1. Pilot test
2. Parallel test
3. Simulation

3.1 Pilot Testing

The pilot concept is very common in engineering where the system is designed and tested on a smaller scale and only after it is successful is the larger scale system designed and implemented. This reduces and localises the danger of disruption to the system.

The pilot system testing concept is similar The system is first tested on a small scale. An example would be the testing of a student scheduling system on one campus before it is applied to all campuses in a multi-campus institution. The pilot situation must, however, be one that can be isolated, such as a campus. One cannot, for example, schedule the students of one department if they are part of a larger system without disrupting the whole system.

Another problem of a pilot study is the problem of scale. The effect of scale is not always linear and must be carefully analysed. At home, the cost of feeding twelve children is less than twelve times the cost of feeding one child. Similarly, with problems of systems, such as information system, the relationships are not linear. The necessary adjustments must be made before projecting the results of scaled system to the actual system. This is not always possible because the relationships involved are not fully known.

3.2 Parallel Run

One solution to the problems of scale is to test the entire system. If this test is to be done before the new system is implemented then this involves running it in parallel to the current system. Such a test is called a Parallel Run or a Dry Run.

The parallel run has the advantage of giving the administrator an opportunity to observe the new system and assure himself that it meets his performance standards before he gives up his current system. The administrator may insist on this approach if he is sceptical of the new system or is sceptical of anything new. The consequences of the system not working may be so great that he may not be willing to take the risk of anything going wrong. Examples of such situations would be the implementation of a new student file and registration and scheduling system or a payroll preparation.

But there is a price for this assurance. Firstly, the parallel run will take more time. Secondly, it will take more manpower. The extra manpower is required not only to check the new system but also to run two systems instead of one. Thirdly, the two parallel systems often result in confusion. All these factors result in considerable organizational strain.

There are some ways of reducing the strain. The administrator can plan for it and hire additional personnel to do the parallel run. The systems designer could also help by designing output that would facilitate checking. Strategies in checking can also reduce the effort required in checking. One approach is to first check totals and check details only if the totals do not match. This is not a very rigorous method of checking because the two compensating errors could go undetected by having unchanged totals. A better approach is to check by taking a statistical sample. If well designed, checking a small sample can be just as efficient as checking the population and will cost much less in time and labour. This subject is discussed further later in this paper.

3.3 Simulation

An excellent example of testing by simulation was the testing of the Apollo craft before it went to the moon. Such testing requires the duplication of the new environment. In the case of information systems, models have been developed that will simulate the new system but these are expensive(1) and not commonly used especially in educational institutions. Hence this subject will not be discussed any further.

4. TESTING CONSIDERATIONS

Other considerations related to testing are the following:

1. Test data preparation
2. Levels of testing
3. Responsibility for testing
4. Test schedule

These will be discussed in turn below.

5. TEST DATA PREPARATION

Test data preparation must answer questions such as the following: How will data be generated? What data will be used? How much data will be required?

In many cases, historical data is available for testing. For example, in converting from a manual to a computer system (without changing the output greatly), the new system can be tested against results of the old system using historical data. Sometimes, however, historical data is not available. For example, this occurs in projects performing functions never done before. In such cases test data must be "invented". This must be done by the administrator and the systems designer. The administrator knows his organization intimately, so that he can anticipate the conceptual problems and exceptions that may arise and these must be tested. The system designer knows the system being tested and can anticipate possible operational problems. Also, he should check for modifications made after the original design stage and ensure their effectiveness.

What data to use and how much data to use is often a problem in sampling. This technique is cheaper and faster and, if well designed, is as significant as the data for the population. The selection of the sample is part of experimental design. Unfortunately, training in this subject is not common. Sackman (1967: 182) points out that:

---

1) For a discussion of simulation of an information system, see McFarlan et al. (1973: 208-232).

> "...programmers are typically not professionally trained in experimental design nor in statistics. The data-processing community has yet to realise that these are key elements in programmer training ....It would be inaccurate and unfair to single out programmers as indifferent or negligent in the application of experimental method. This charge may be levied against most managers, designers, and users as well. There exists an experimental lag in contemporary man-machine digital systems. Relatively few people working with such systems have received formal training or applied experience in empirical experimental techniques, particularly techniques for dealing with both men and machines in computerised settings."

Experimental design is also relevant to system design. Again to quote Sackman (1967: 185):

> "If system design is not specified in the form of performance hypotheses, and if system design is not tested by empirical experiments, then system development will remain at the level of crude technology and will not be able to advance to the status of applied science."

Once the test data collection strategy and other test planning strategies are selected (activity 400-480 in Figure 1), the test data can be collected (activity 480-700 in Figure 1). Testing can then commence and can be performed at different levels. These levels of testing will now be discussed.

6. LEVELS OF TESTING

There are many levels at which a system can be tested. These are:

1. Component testing
2. Functional testing
3. Subsystem testing
4. System testing

6.1 Component Testing

A component test is the checking of the separate components of the system. These components may be a piece of equipment, the performance of an individual operator, a form, a procedure, or a program.

An example of system component test is desk checking of a computer program. In this method, the programmer checks the program solution at his desk by methodically checking each output result manually, and checking especially for omission and possible misinterpretations.

Component testing is not always easy because of the difficulty of isolating a "component". A person's work may be so related to his colleagues' that testing of an individual's work is not meaningful. The component would then be the aggregate work of the group of workers.

In equipment, the problems are the same. One need not test every individual part of a machine, such as a keypunch, but the operation of the entire machine.

Component testing is important because a component is a basic element of a system. Identifying errors and problems in a component is easier than isolating a problem in a group of interrelated components. Also, the earlier the problem is isolated and detected in the testing process, the less the effort required to correct it.

6.2 Functional Testing

Functional testing is at a higher level of aggregation than component testing, since it tests a functional subsystem which may consist of many related components such as men, machines, and procedures. This level of testing should be done after the corresponding components are tested and found satisfactory.

6.3 Subsystem Testing

A subsystem would be an aggregate of related functions, such as an input subsystem. This subsystem includes functional activities of input procedures, input preparation, validity checking, and procedures for correcting errors. Another example would be an update subsystem, which is also a grouping of related functions that are isolated enough to enable independent testing. Testing the subsystem would test the effect of the interrelationships of functions tested individually earlier.

6.4 System Testing

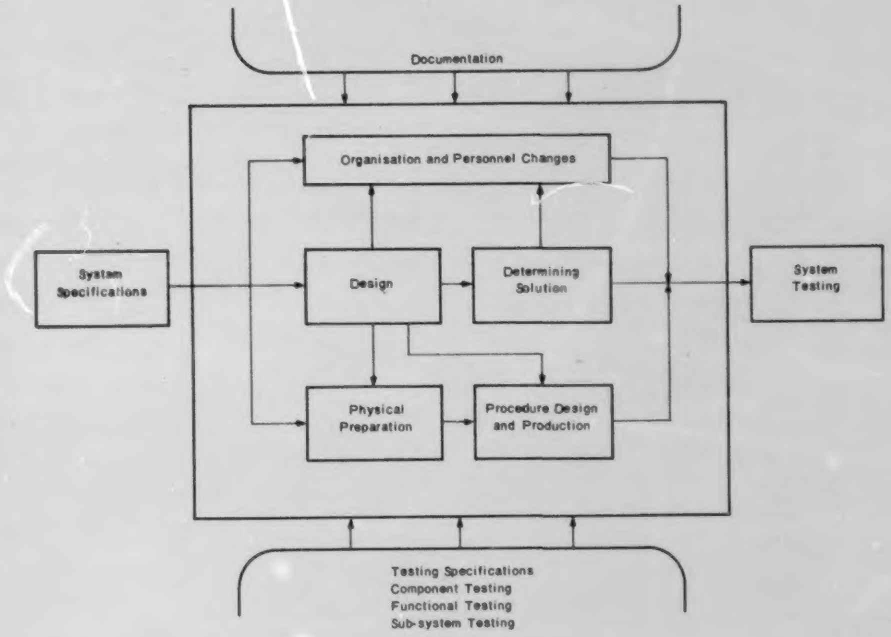
System testing is the test of the final aggregation of all the subsystems. It checks for overall results. The successful testing of the whole system completes the testing process.

The testing by levels discussed above goes from the lowest level of detail to the highest level of generality and aggregation. This is the opposite of the early stages of the development process, where the aggregate and general problem definition was followed by the designing of the details. Once the details are designed, the components, functions and subsystems are tested in stages up to the aggregate whole, or the overall system as originally envisaged. This closes the loop from conception of the system to its realisation.

The relationship of the testing to other development activities is shown in a block diagram in Figure 2. It clearly identifies the

Figure 2

TESTING PROCESS AND ITS RELATIONSHIP WITH OTHER STAGES OF SYSTEM DEVELOPMENT


Documentation
Organisation and Personnel Changes
System Specifications
Design
Determining Solution
System Testing
Physical Preparation
Procedure Design and Production
Testing Specifications
Component Testing
Functional Testing
Sub-system Testing

continuous feature of the testing process (as well as documentation). These are performed throughout the development process as soon as is feasible.

7. TESTING RESPONSIBILITY

A classification of testing methods that helps identify the responsibility of testing is the distinction between outside and inside tests. Van Court Hare (1964: 136) makes the distinction as follows:

> " In the outside tests, the system is viewed as a black box and the relation between inputs to the box and outputs from it, that is, the system response is sought for a wide range of input conditions.
>
> With the inside tests, the structure of the system is examined for completeness, consistency, and correctness of transformation at each element.
>
> Frequently, inside and outside tests are combined, so that a knowledge of system structure and internal operation can suggest critical response tests, or a sequence of input conditions that will isolate defective or malfunctioning elements."

The two tests are further explained by Van Court Hare (1964: 136) in the table shown below.

Figure 3

OUTSIDE AND INSIDE TESTS

Systems Analysis
Test or Questions

Outside	Inside
Response Tests	**Structural Tests**
Frequency	Omissions
Overload	Compatibility
Stability	Coupling Effects
Fidelity to Specifications	Alternates
Reason	**Reason**
Prediction	Check definition
Limits of operation	Understand structure
	Troubleshooting
	Improvement

The above classification helps to distinguish the responsibilities of the user and systems personnel in regard to testing. The "outside" testing is largely the user's responsibility. He must help the systems personnel in designing the inputs and must check the outputs for each combination of inputs. The "inside" tests are exclusively the responsibility of systems personnel, mostly the programmers.

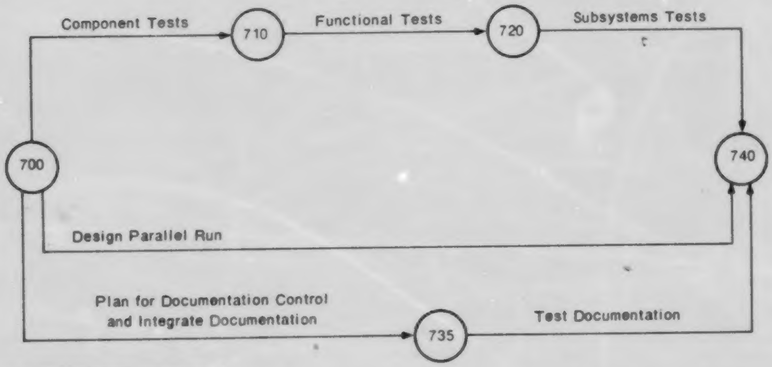
Some jobs must be tested by the person performing the job. An example would be a computer program. The programmer knows the problem very intimately and he can spot the errors readily. Besides, it is often his responsibility to debug(1) his program (remove the errors). However, the program should also be tested by someone else to ensure objectivity, and also to detect errors in assumptions which the programmer cannot test because he has overlooked them to start with. In the case of programming, the program can also be tested by the systems designer who knows the system and would anticipate some of the problems. Again, he is involved and may not be objective. Testing by a systems analyst completely disconnected with the project would eliminate this problem. In cases of programs involving funds or an accounting procedure, an independent auditor can well perform the test function.

8. TEST SCHEDULE

An important part of planning a test is the scheduling of the different tests: their sequence and their timing must be determined. Some tests can be done in sequence, others in parallel and sequential tests. One schedule, a sequential set of tests, is shown below.

Figure 4

PARTIAL NETWORK ON EARLY TESTING ACTIVITIES



---

1) This term originates from the early days of computers that used vacuum tubes. One such computer gave false results. The programmers spent many days trying to trace the error but with no success. Then, by accident, an operator found a bug on a contact that was causing the machine to malfunction. The "bug" was removed and the program worked perfectly.

9. CONVERSION AND RELATED ACTIVITIES

The component, functional and subsystem test activities in Figure 4 have been discussed earlier. In parallel to these tests there is the activity of designing the parallel run (activity 700-740 in Figure 4). (It is assumed that a parallel run strategy has been chosen. This and other test strategies were discussed earlier.) The design would include the assignment of personnel to perform the jobs; identification of control points and controls to be used in testing; recording of results; and finally, the setting up of liaison between the user and systems personnel for the purpose of implementing the parallel run. This is a stage where the user and systems personnel must work very closely with each other, since it can be a time of considerable organizational strain.

By this time in the development process, most documentation should have been completed in each stage such as forms and procedures, making other physical preparations, the rewriting of programs, or even the restating of system requirements. This recycling must be carefully controlled and formalised. Procedures must be established for recording the modifications desired and accomplished as well as procedures for channelling the feedback. This closing of the loop is not only necessary during the development stage but is important during actual operations where problems may arise that were not detected in the testing phase.

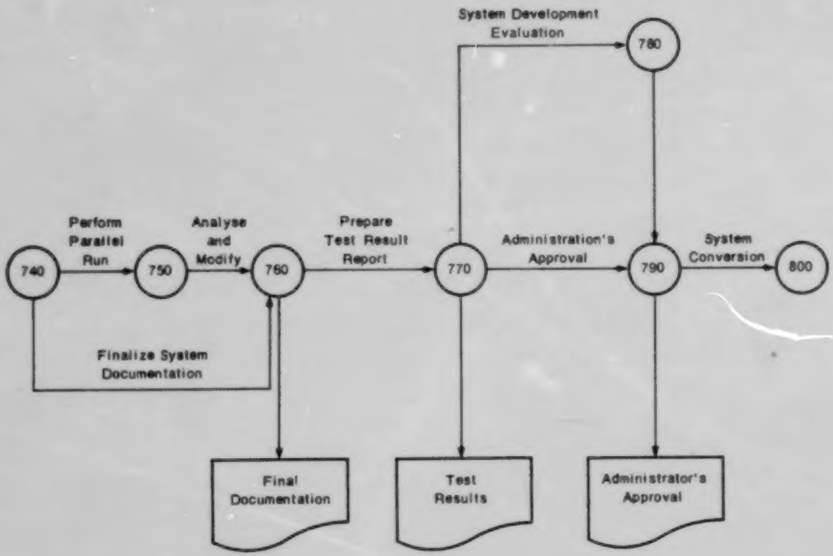
The recycling of the testing process requires iterations from testing to redesigning, recollecting of data for testing, recomputing the solution, and retesting until such time that the evaluation is satisfactory. This recycling is depicted in Figure 6 where n possible cycles are shown (Bross, 1953). In complex systems, the value of n may be 3 or 4. If it is much more, then either the user did not adequately specify his needs to start with: the systems personnel did not do a careful design job; or the development process was not sufficiently planned and controlled. The value of n is seldom 1 even with capable personnel. Administrators must recognize this possibility and be prepared for it.

If the number of recycles is large, then there may be a slippage of the time schedule as well as an increase in resources required. This must be minimised wherever possible. Heany (1968: 124) points out:

> "Some adaptions in design specifications are possible and desirable. Wherever possible the designer tries to accommodate the users. At the same time he must insist that major alterations in information requirements be formally introduced. If the user wants to halt systems work and reopen questions decided earlier in system development, it is only proper to retitle the

Figure 5

PARTIAL NETWORK ON FINAL TESTING AND CONVERSION ACTIVITIES


System Development
Evaluation
780
Perform
Parallel
Run
Analyse
and
Modify
Prepare
Test Result
Report
Administration's
Approval
System
Conversion
740
750
760
770
790
800
Finalize System
Documentation
Final
Documentation
Test
Results
Administrator's
Approval

study. Drifting information requirements can, if not labeled, give the information systems component an undeserved reputation for never meeting its schedule or living within its resource commitments."

Following the satisfactory completion of testing and systems documentation, the test results are presented to the administrators (or managers). The formal approval of administrators is very important (activity 770-790 in Figure 5). It is an opportunity for the administrators to ensure that the system meets their goals and objectives and satisfies the systems requirements they set. This is an activity that the systems personnel sometimes insist upon, since it formalises user's acceptance and minimises the possibility of the user complaining later that he did not get what he wanted.

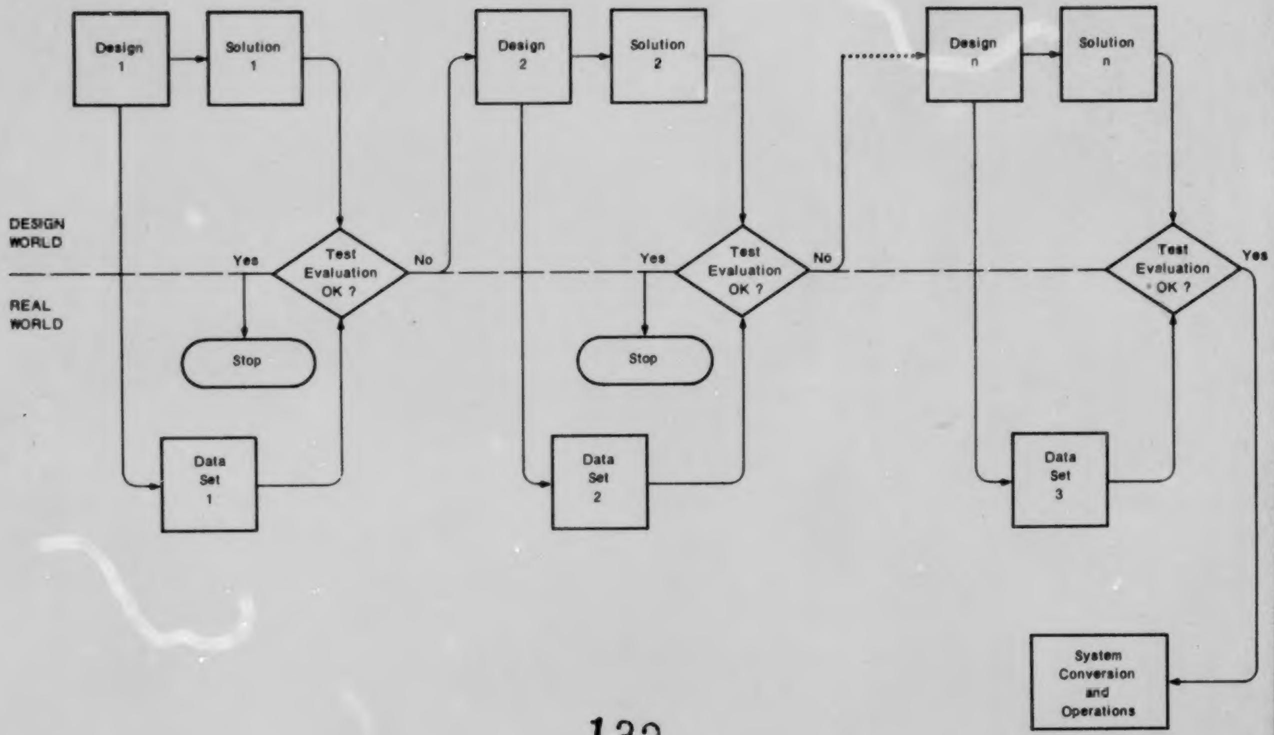
Another activity to be performed is the evaluation of the systems development process (activity 770-780 in Figure 5). Such an evaluation can be done in different groups: at the technical level among systems personnel; between the users; and with the users and systems personnel. In all cases the project management personnel must be involved. The mistakes made should be identified and analysed as to why they were made and how they could have been avoided. The purpose of such an analysis is not to blame or persecute anyone, nor to exchange accusations and recriminations, but to benefit from one's mistakes. This is an important part of the learning process. It must be recognised that the nature of the development process is such that mistakes will be made. By recognising and analysing them they can hopefully be avoided in the future. It is not only important that the project is completed, but that it is completed efficiently and effectively. Administrators should emphasize these aspects of the project. They must assess not only the benefits of the system but also the costs incurred. They should question the approaches taken, the need for recycling, the problem of liaison and communication, the effective use of manpower, and the slippages in schedule (if any). A critical review of this type may lead to improved procedures and approaches to be followed in future projects.

The final step in the development process is the cut-over or conversion from the old system to the new system (activity 790-800 in Figure 5). Part of the phasing-out process is the withdrawal of the old documentation referred to earlier. Other items that must be withdrawn and retrieved are forms and files that are outdated. Equipment no longer required must also be removed. Any unnecessary delay in the phasing-out of old equipment can result in additional costs on rentals.

Sometimes part of the phasing-out process would start much earlier in the development process, even in the design stage. An example would be the retirement of old forms. This can be done once

Figure 6

RECYCLING OF TESTING PROCESS


Design 1
Solution 1
Design 2
Solution 2
Design n
Solution n
DESIGN WORLD
REAL WORLD
Yes
Test Evaluation OK ?
No
Yes
Test Evaluation OK ?
No
Test Evaluation OK ?
Yes
Stop
Stop
Data Set 1
Data Set 2
Data Set 3
System Conversion and Operations

the new forms are designed. If the new forms incorporate the data required previously, then the new forms can be used in the old system, resulting in an early conversion. This reduces the pressures at the cut-over period when other conversions are taking place.

The phasing-in of the new system must be carefully planned. Reference was made to the distribution of manuals to relevant personnel. These personnel and others that will operate the system must be given their final orientation concerning the new forms, procedures and equipment. They must also be briefed on a schedule of the cut-over dates and the new organizational relationships (if any).

There is often a tendency among user personnel to delay the phasing-out of the old system. They prefer the continuation of the parallel run because of the sense of security associated with the older system. This must be discouraged. Once the new system has achieved its desired level of performance, the old system must be cut-off. The swifter the phase-out, the less the reliance on the old system and the less the organizational disruption and strain. Chapin (1955: 165) points out:

> "The organisation is under considerable strain during the period of the parallel operation and if it is long continued, organisational breakdowns tend to occur. Again, unless cut off promptly, the old system becomes entrenched at the very time it is redundant. Once a precedent has been established, it may be hard to root out the remnants of the old system later. The new system, if it is ever to earn the confidence of personnel in the organisation, must stand on its own feet without a backstop. Parallel operations must be conducted only long enough to establish confidence; continuing them too long is a sign of weakness in the new system."

Approaches to the cut-over from an old system to a new one are also discussed by Orlicky (1969: x 149-150):

> "A very effective conversion method is the so-called phase-in/phase-out, whereby the old system is permitted gradually to expire, while the new system is phased-in...But in many cases these methods will not be suitable, and conversion must take place abruptly, overnight. This method is popularly known as burning the bridges (also "cold turkey")... It provides a powerful incentive for all system user personnel to make the new system work."

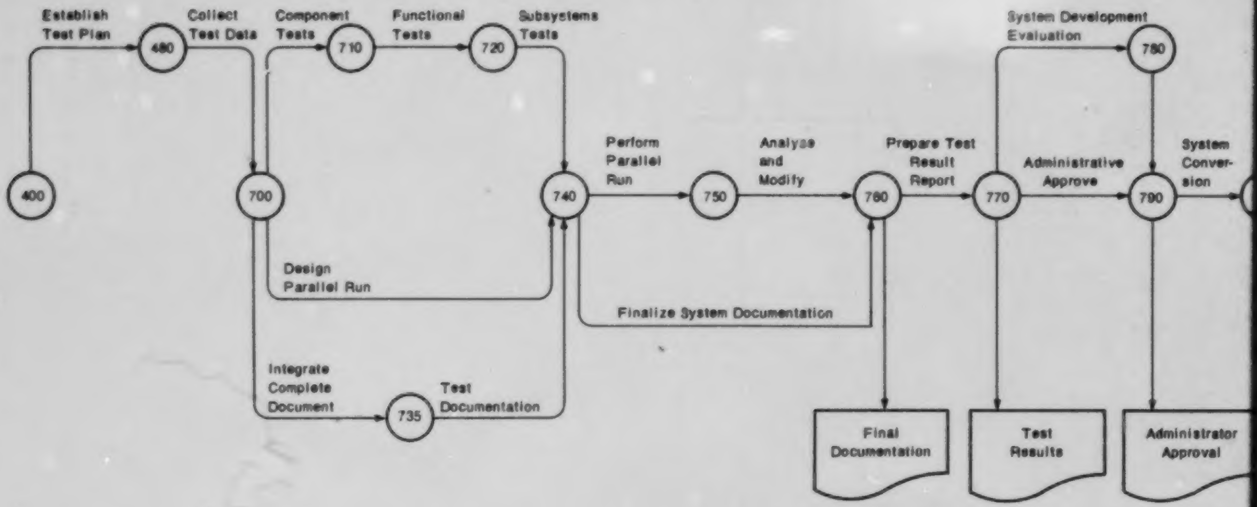
10. SUMMARY AND CONCLUSIONS

The activities of testing and conversion are shown in Figure 7.

The important points concerning the testing and conversion stages of system development are as follows:

Figure 7

NETWORK FOR TESTING AND CONVERSION ACTIVITIES


Establish Test Plan
Collect Test Data
Component Tests
Functional Tests
Subsystems Tests
System Development Evaluation
400
480
700
710
720
740
750
760
770
780
790
Perform Parallel Run
Analyze and Modify
Prepare Test Result Report
Administrative Approve
System Conversion
Design Parallel Run
Finalize System Documentation
Integrate Complete Document
735
Test Documentation
Final Documentation
Test Results
Administrator Approval

1. Testing is a comparison of expected operational performance with desired performance as stated in the systems specification stage.
2. The desired performance and the testing specifications must be determined by the user and the systems personnel.
3. The systems design should facilitate the testing stage by enabling the isolation and detection of errors and problems.
4. The systems personnel typically design the testing methods and testing experiment such that the system is given a thorough trial for both normal and abnormal conditions.
5. The testing stage must be carefully planned. Equipment time for testing must be scheduled to avoid delays in getting test results.
6. The user must participate actively in the testing stage - especially at the operational and supervisory level. They must insist that the desired specifications are met to their satisfaction.
7. The testing stage provides an excellent opportunity for training personnel that will eventually be responsible for the operations of the system.
8. Testing should be done at different levels of component test, functional test, subsystem test and then the system test. The testing should also be done - whenever possible - at increasing levels; low load to normal load to peak load; small subsystems to large and then larger subsystems; and simple to complex subsystems.
9. The testing should encompass all elements of the system; the hardware, input, output, interfaces, feedback, formats, codes, programs, training and procedures for both normal and exceptional conditions.
10. The performance must be tested not only for quality but also for its timing, using the safety factors specified in the system specification stage.
11. In large systems, the testing could be done on a small scale and is referred to as a pilot test. Allowances must be made in predicting operational performance changes in scale between the pilot system and the actual system.
12. Parallel operations should be performed on all complex and crucial systems changes. Such a testing technique requires additional effort and could result in considerable organizational strain unless it is carefully planned and the personnel well briefed on its need.
13. The period of parallel operations should not be extended beyond what is necessary for adequate testing. Undue delays in conversion would result in additional organizational strain.

14. Interested personnel should be carefully and regularly informed of the status of conversion and project completion. Silence and absence of information could start rumors resulting in low morale.

15. The duration of testing and conversion to an operational stage is characterised by Chapin (1955: 359) as follows:
    1. It is a period of strain and disruption
    2. Higher than normal work load
    3. Things rarely proceed as they should
    4. Unforeseen difficulties crop up and must be disposed of, often under great pressure of time
    5. Mistakes are made, cost mounts up rapidly as overtime is paid and work is redone
    6. Last-minute changes resulting in reprogramming
    7. Normal service is unavoidably affected.

16. Some of the above problems can be overcome if they are understood and anticipated.

REFERENCES

Bross, Z.D.J., (1953) Design for Decisions, (New York: The MacMillan Company).

Chapin, N., (1955) An Introduction to Automatic Computers, (2nd Ed.), (New York: D. Van Nostrand Company, Inc.).

Heany, D.F., (1968) Development of Information Systems, (New York: The Ronald Press Company).

McFarlan, F.W., P.L. Nolan and D.P. Norton, (1973) Information System Administration (New York: Holt, Rinehart, and Winston, Inc.).

Orlicky, J., (1969) The Successful Computer System, (New York: McGraw-Hill Book Company).

Sackman, H., (1967) Computers System Science and Evolving Society, (New York: John Wiley & Sons, Inc.).

Van Court Hare, (1964) "Systems Analysis" in D. Hertz and R. Eddison, II, eds. Progress in Operations Research, (New York: John Wiley & Sons, Inc.).

Part Three

OPERATING AN INFORMATION SYSTEM

The previous Part of the book discussed the initial development of an information system to the point that it can be converted and be operational. But to remain operational, the system must be evaluated, maintained, controlled and organized. These are the topics of this part of the book.

Most information systems operate in a changing environment - both organizational environment and the technological environment. Therefore the information system must be regularly evaluated to ensure that it is relevant and performing according to expectations and specifications. If not, then the system must be modified - a process referred to as maintenance. This will require redevelopment of the system and a recycling of the entire or part of the development process discussed in Part Two. The maintenance and evaluation is the subject of Chapter 7.

The evaluation of a system may be satisfactory, that is, the system may operate within expected levels of operation and errors and yet it is necessary to control the system for quality of information and to ensure that these errors are minimised. This is the subject of Chapter 8.

For the evaluation, maintenance, control and the operation of the system, it is necessary to organize the information system. Much of the organization is technical but some of it is of interest to the manager and important to the institution. This is the subject of Chapter 9. In this chapter, numerous issues related to institutional management will be identified. One important issue is the justification of the information system; its costs and its benefits. These topics are discussed under the Economics of Information, the topic of Chapter 10.

Chapter 7

EVALUATION AND MAINTENANCE OF AN INFORMATION SYSTEM

by

K.M. Hussain

7

CONTENTS

1. Introduction 144
2. Nature of Evaluation and Maintenance 144
3. Evaluation and Maintenance Recycle 144
4. The Necessity of Maintenance 147
    4.1 Institutional Needs 147
    4.2 Data Processing and In-House Needs 147
    4.3 Changes in Exogenous Variables 148
5. Evaluation and Maintenance Procedures 149
6. Summary and Conclusions 149

References 150

Selected Annotated Bibliography 339

1. INTRODUCTION

In this chapter the nature of evaluation and maintenance is examined and their interrelationship explained. Further, the factors that make maintenance necessary are identified and procedures for maintenance are discussed.

2. NATURE OF EVALUATION AND MAINTENANCE

Evaluation is the comparison of actual performance with specified performance. If there is a divergency and if this is beyond the level of tolerance set by the specifications, then changes must be made to reduce the divergency such that the system can continue to meet its objectives. The process of implementing the necessary changes is referred to as maintenance.

It should be emphasized here that if the objectives of the system should change resulting in major changes to the design of the system, then the necessary modifications are not typically considered to be maintenance and a new system must be developed. Changes resulting from maintenance are only those that are concerned with ensuring that the system is performing according to current specifications. The term maintenance also covers a wide range of other activities such as repairing broken-down equipment and changes to a file. These are beyond the scope of this paper and will not be discussed.

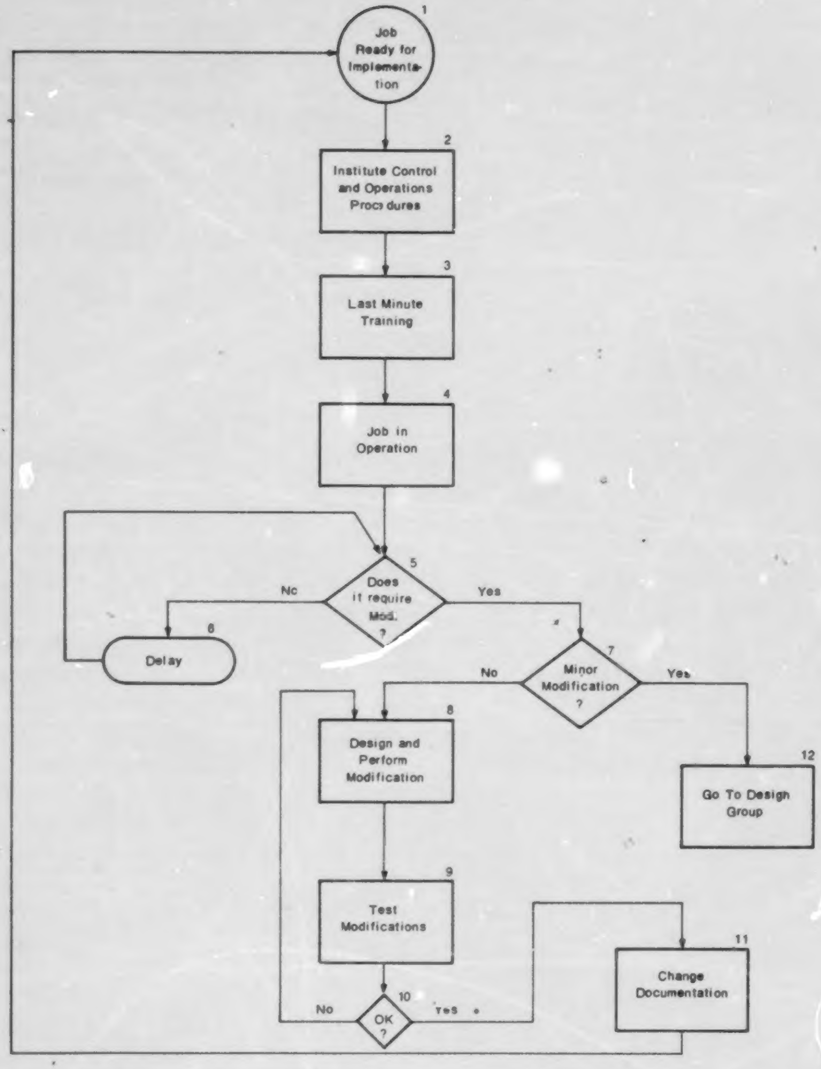
3. EVALUATION AND MAINTENANCE RECYCLE

The process of evaluation and maintenance consists of a number of activities and logical decision points. These are shown in the flowchart in Figure 1. Also shown are the related activities connecting the point where the system is originally developed and ready for implementation to the process of evaluation and maintenance.

Referring to Figure 1, the system is ready for implementation (box 1) and is followed by control and operations procedures being instituted (box 2). Last minute training is completed (box 3) and then the job is in operation (box 4). Periodically, an evaluation is performed (box 5). If the evaluation shows that no modifications are necessary ("No" exit of box 5) then the system operates for a predetermined period of delay (box 6) and another evaluation is performed (box 5). If a modification is found to be necessary ("Yes"

Figure 1

FLOWCHART


1
Job Ready for Implementation
2
Institute Control and Operations Procedures
3
Last Minute Training
4
Job in Operation
5
Does it require Mod. ?
No
Yes
6
Delay
7
Minor Modification ?
No
Yes
8
Design and Perform Modification
12
Go To Design Group
9
Test Modifications
11
Change Documentation
10
OK ?
No
Yes

exit of box 5), the magnitude of the change must be determined (box 7). If the change required is a minor one ("Yes" exit of box 7) (such as a programming oversight) and does not affect the overall design of the system, then the change is designed and made (box 8); the system tested for the change (box 9) and if successful ("Yes" exit from box 10) then the system is documented for its modification (box 11). The job is then ready for recycling at box 1 in Figure 1. If the result of the test of the modification is unsatisfactory ("No" exit for box 10) then the modification must be redesigned and reimplemented (return to boxes 8 and 9).

In checking the nature of the modification required, it may be found to be a major one ("No" exit of box 7). This will require an alteration to the original system design, and hence the design-group must be consulted (box 12). The group will then conduct a feasibility study and, if successful, the development cycle is then initiated (not shown in Figure 1).

For evaluating current systems, Kirzer (1972: 30) has presented a list of questions that should be asked:

1. Have changing management objectives and problems been effectively translated into revised systems requirements and incorporated as modifications to the system?
2. Have anticipated benefits been fully achieved?
3. Have original system objectives been undesirably compromised in an attempt to satisfy other objectives the system was not designed to encompass?
4. Does a favourable cost versus benefits trade-off exist? (An attempt to establish the value of information in the hands of the user is useful.)
5. Is there additional manual processing of data using computer output which would be more cost effectively accomplished by expanding the system's programs?
6. Is all system output used?
7. Are meaningful data reduction and exception reporting techniques employed to produce digestible, usable decision-oriented information?
8. Do adequate procedures exist governing preparation and control of input data?
9. Does the system employ sufficient data edits/audits?
10. What is the general level of confidence placed in system output by the users?
11. Is the precision, as well as the accuracy, of the system's results understood by the users?
12. Are processing turn-around times satisfactory?
13. Are processing delays frequent in occurrence?

14. Are obsolete or unused data retained in system files and unnecessarily processed?
15. Is processing logic straightforward, easily understood, and properly documented? Or, are routines unnecessarily complicated?
16. Do adequate processing operation procedures exist?

Consideration of benefits and benefit-cost ratios are discussed in a separate chapter and will not be discussed further. There are, however, many references on this subject in the annotated bibliography.

The flowchart in Figure 1 is an infinite loop because of the continuous nature of evaluation and maintenance. The importance of this cycle cannot be overemphasized. It is essential to all information systems which operate in a changing environment.

4. THE NECESSITY OF MAINTENANCE

There are many variables in an information system which may change making maintenance necessary. These may be the needs of the institution, the needs of the data processing organization, or variables from outside the system. These variables will be discussed in turn below.

4.1 Institutional Needs

The need for maintenance can arise because of structural or personnel changes within the institution. For example, a change in a high level decision-maker may result in changes to the content and format of reports produced for him. Alternatively he can initiate a policy change which will affect the system, such as a change in fringe benefits for certain employees of the institution, or changes in the decision rules on probation of students. Such changes will require changes in related programs resulting in maintenance work.

Maintenance changes may also result from institutional growth. Consider, in illustration, an increasing student enrollment. This will involve enlarging files, expanding or perhaps completely changing coding systems, finding faster and better techniques for processing, or perhaps acquiring faster equipment for such activities like data gathering and data preparation. These resulting activities are part of maintenance.

4.2 Data Processing and In-House Needs

In a data processing department, there are many changes which commonly take place. One example of these is a change in codes resulting from the need to link a related subsystem. This requires that manuals related to the changed codes be updated, and that computer programs which use such codes be modified.

Equipment changes are not uncommon in the computerised information system. Their change leads to changes in compilers, program packages, and application programs. Even with compilers that have been used for a long time, modifications are sometimes necessary resulting in changes necessary to application programs using the compiler.

Sometimes errors are made in the original system design. For example, a systems analyst could have made a wrong assumption or left out a decision rule or the administrator may have omitted or mis-stated one of his needs. Also, inefficiencies are often discovered in the system or a new and better technique of solution is discovered. In all these cases, maintenance work is required.

4.3 Changes in Exogenous Variables

Maintenance can be caused from the effects of variables outside the system. As an example, consider the federal government changing its income tax rates. This would call for changes to the information subsystem that computes payrolls. Other examples of external influences are laws, government regulations and industry standards and practices.

Because of the many reasons for maintenance discussed above, the amount of time spent on maintenance can be quite large. The table in Figure 2 shows the results of studies made on the amount of time spent in various systems activities by applications programmers and by systems programmers. Both groups of programmers spend a considerable time on maintenance.

Figure 2

PERCENTAGE OF TIME SPENT IN VARIOUS SYSTEMS ACTIVITIES

Systems Activity	Per cent of time spent in Applications Programming	Per cent of time spent in Systems Programming
1. Problem Analysis	25	30
2. Flowcharting	10	05
3. Coding	20	15
4. Debugging	20	15
5. Documentation	10	10
6. Maintenance	15	25

5. EVALUATION AND MAINTENANCE PROCEDURES

Maintenance is not only costly in financial resources but is also consuming in time and disruptive to the organization. To minimise the costs of maintenance, procedures must be set up and enforced for its orderly organization and control.

Some procedures are implied in the flowchart shown in Figure 1. Others not implied concern the initiation of a request for maintenance and then the decision on whether or not the requested maintenance is performed. These procedures are discussed below.

The need for maintenance would typically be channeled through either the user, administrator or the systems analyst. Needs arising because of external variables such as changes in tax laws should be perceived by the user and, similarly, changes arising because of processing hardware or systems software should be perceived by the systems analyst. These needs must be anticipated as early as possible in order to allow for an orderly process of evaluation and development. They should be presented through prescribed channels for evaluation of such needs.

Typically, the maintenance is a service function performed within constrained resources. The allocation of these resources is often the function of a standing committee that is representative of the main users and the systems department. They must ask such questions as: Is this a problem in maintenance? If so, is it feasible? What priority should it have compared to other needs? Once these questions are answered and if a maintenance request is approved, it is then scheduled and must go through the basic process of development such as discussed in the previous part of the book. In many cases some stages in development may be trivial or not necessary but conceptually they should not be ignored.

6. SUMMARY AND CONCLUSIONS

After an information system has been implemented and is operating, it will not always continue to operate efficiently and effectively without changes being made on it. Variables both from within and outside the system will necessitate changes in order that the system can continue to meet its objectives. Making such changes to the system is referred to as maintenance.

Closely related to maintenance is evaluation. Evaluation of the system must be performed in order to determine when maintenance changes may be necessary. The final decision on maintenance is typically made by a committee that is representative of all users.

Maintenance is not only resource consuming but is also unpopular with system personnel because it is not considered too "creative".

It is also very often disruptive to the organization. Hence, once maintenance work is decided upon and scheduled, it should not be changed unless absolutely necessary. A situation should be avoided where the user is constantly changing his mind about what maintenance work needs to be done. Following prescribed procedures for evaluation and maintenance will help avoid unnecessary maintenance which would result in a waste of time, money and human effort.

Related to the problem of evaluation and maintenance of an information system is the control of the quality of information generated by the information system. The information in terms of its accuracy and validity must be within the tolerance levels specified by management. The specifications of these levels should have been specified in the design stage. The monitoring of these tolerance levels and procedures necessary to maintain the desired level of quality of information is the subject of the next chapter.

REFERENCES

Kirzer, J.G., (1972) "A Model of System Design", Journal of Systems Management. (October) pp.24-30. Also reprinted in Sanders (1974: 287-301).

Sanders, Donald H., (1974) Computers and Management in a Changing Society, 2nd edition, (New York: McGraw-Hill Book Company).

Chapter 8

QUALITY CONTROL OF INFORMATION (1)

by

K.M. Hussain, New Mexico
State University, U.S.A.
and
C.R. Thomas, College and
University Systems Exchange,
Colorado, U.S.A.

CONTENTS

1. Introduction 152
2. Quality Control Points 152
    2.1 Documentation 154
    2.2 Forms 155
    2.3 Data Collection 156
    2.4 Data Preparation 158
    2.5 Programmed Detection of Errors 160
    2.6 Operations 161
    2.7 Data Files 164
    2.8 Programming 164
    2.9 Processing 167
    2.10 Output 167
    2.11 Administrative Responsibility 168
3. Summary and Conclusions 170

References 170

Annotated Bibliography 341

---

1) This paper was originally written for availability in the College and University Document Exchange Library, 737 Twenty-ninth Street, Boulder, Colorado, U.S.A. It has been adapted for the IMHE seminar on "Information Systems for Institutions of Higher Education" and the European audience.

1. INTRODUCTION

All information systems are susceptible to errors. This is especially true of computerised systems. As Eckman and Froberg (1965) put it: "Never before in history has it been possible to produce so many errors in so short a time". Most of these errors occur because of carelessness, poor systems design, bad programming, inadequate procedures, or human problems. Some errors are intentional, as in the case of fraud; some are predictable, as those caused by poor planning; and others are neither intentional nor predictable, such as damage from fire, flood, or other catastrophes.

This chapter will describe some of the errors and some methods for eliminating or minimizing them. It will identify points in the processing cycle where errors occur, describe the probable causes, and discuss the techniques for controlling them. This discussion does not include highly technical details of operational efficiency or performance of hardware and software; such matters are of more direct concern to the management of computer operations than to the management of design, production, or general functions.

2. QUALITY CONTROL POINTS

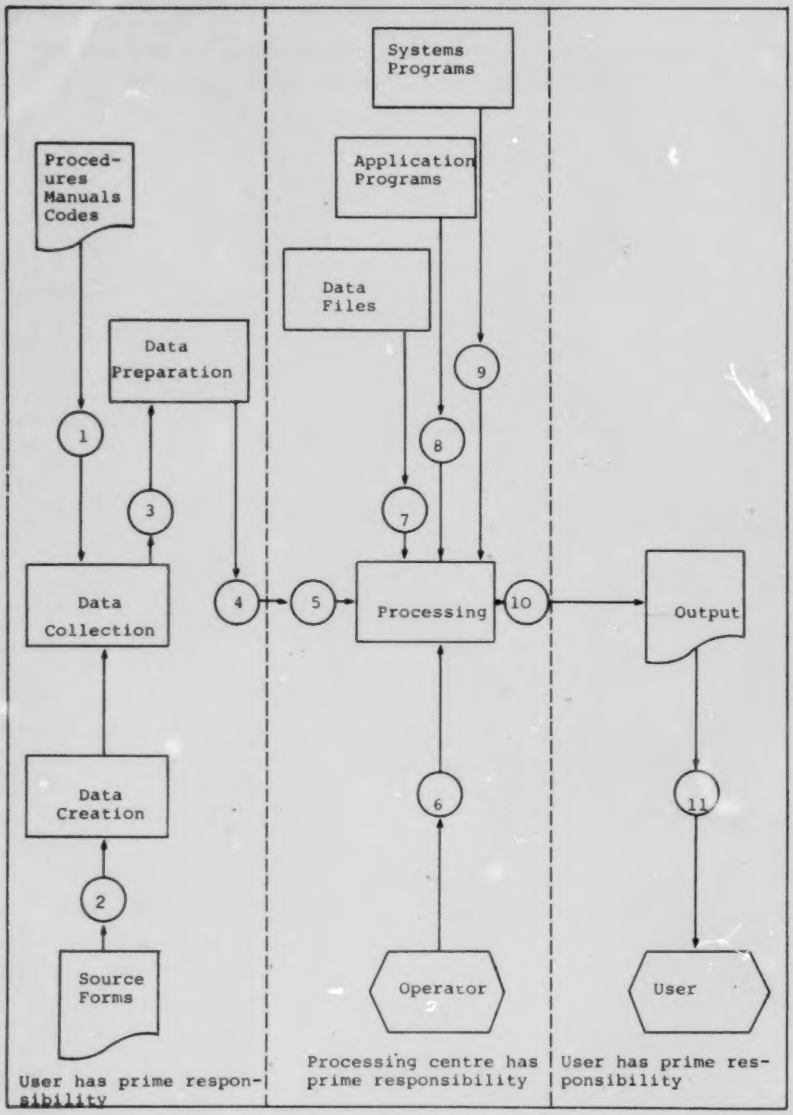
The steps in generalised processing from input to output are shown in Figure 1, with numbered circles indicating the points where errors may occur. These are grouped within the vertical lines of Figure 1 to identify the typical areas of responsibility. The groupings may vary from institution to institution. For instance, data preparation (error point 4) may be the responsibility of the user as shown, or it may be assigned to the data-processing centre. Either case may be appropriate for a particular information system.

Each of the points of Figure 1 will be discussed in this chapter. The numbering of the subdivisions of section 2 corresponds to the numbering of the points. The errors that may occur at the point are described and corrective considerations are discussed. A tabular summary for each control point names the potential errors, gives examples, and suggests methods for detection, reduction, or elimination.

System design flaws of a general nature may cause errors at many points in processing, and will be discussed.

Figure 1

STAGES OF PROCESSING AND QUALITY CONTROL POINTS


Systems Programs
Procedures Manuals Codes
Application Programs
Data Files
Data Preparation
9
1
8
3
7
Data Collection
4
5
Processing
10
Output
Data Creation
6
11
2
Source Forms
Operator
User
User has prime responsibility
Processing centre has prime responsibility
User has prime responsibility

2.1 Documentation

Errors often arise from poor documentation or the misuse of proper documentation, especially procedure and code manuals. Documentation errors that frequently occur are shown in Figure 2. The first involves a poorly-written manual that leads to misunderstanding. This problem can be minimised by carefully-designed manuals that are tested thoroughly before their actual use in the system.

Figure 2

SUMMARY OF ERRORS FROM USE OF MANUALS

Type of Error	Example	Solution
Poorly-written manual	Wrong procedure or code used because manual not clear Manual is incorrect, incomplete, or ambiguous	Better manual . Design . Test
Wrong manual	Incomplete or outdated manual Unauthorised access to manuals Unauthorised copying of manuals	Make sure every manual is updated Control distribution of manuals

Manuals must be written in the appropriate level of language in terms of definitions and vocabulary with simplicity and clarity. The real user of the manual must receive first consideration. A manual that is adequate and understandable to a systems analyst is not necessarily adequate and understandable to a member of the clerical staff. A similar rule applies to the testing of documentation: all manuals should be tested by the proposed users rather than the designers.

Standardization of the techniques and forms for documentation can significantly improve its effectiveness as an aid to all concerned with development, implementation, and operation of data processing systems. The professionalism of a data processing staff can be measured by the quality of the documentation that it produces.

Another error can occur through lack of control in the distribution of documentation. A common problem is an inadequate means for distributing changes applicable to existing manuals, leading to the use of outdated, and incorrect material.

Another problem arises when an unauthorised person has a manual, or an unauthorised copy has been made, resulting in the use of a copy that is not on the list for distribution of changes. These problems can be minimised by a carefully controlled system for distributing manuals and updates; this system should include distribution dates for manuals and all changes with appropriate references to the locations of the master copies of the documentation. There should also be policies and control concerning duplicating and copying.

There are many offices and agencies in an institution of higher education that are vitally concerned with its codes. It is desirable to designate one office as the "keeper of the code". This office must co-ordinate assignment of codes according to the needs of various users, and must be responsible for the publication, distribution, and maintenance of a "uniform code manual" for all codes used in administrative systems. Such co-ordination can eliminate redundant coding schemes and wasted effort while ensuring the existence of the codes required for proper information.

2.2 Forms

A common method of collecting data for an information-processing system is through the medium of forms. In many cases where the correct completion of forms is left to the individuals using the forms, it may be impossible to ensure error-free completion. Errors can be minimised, however, by having well-designed forms with instructions that are easy to understand, unambiguous, and complete.

In many cases, the directions for filling out a form are either inappropriately located on the form or are printed on separate documents that may not accompany the form. Some data elements lend themselves to a self-explanatory design. Other data elements can be designed in a format that will be familiar to the user; for instance, the format for a Social Security number shown in Figure 3 is preferable to an unformatted space for the nine-digit number.

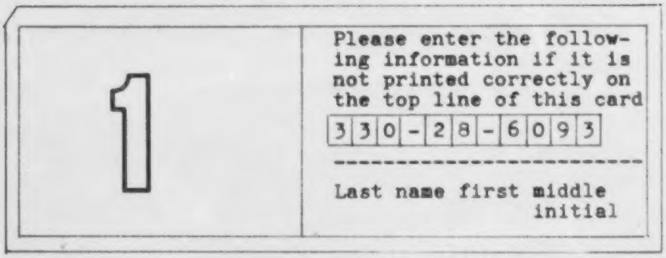
In the design stage, all forms should be tested with a sample group of the people who will be required to complete the forms in actual practice. A team of systems analysts is not an adequate test group for forms to be filled out by students, although the analysts can help in the early stages of testing.

Errors can also result from a form being filled out by an unauthorised person, as in the case of a student having a friend go through the registration process for him: the substitute probably will not know all of the necessary information, and may guess at or omit vital data. This situation is difficult to control, but the problem can be minimised by appropriate control of the distribution and collection of forms and by use of "turnaround" documents on which identification and other vital data have already been entered.

Poor motivation can be a large factor in causing errors. A person who is not properly motivated may not bother to pay attention to correctly filling out a form. Instructions can play a part in eliminating this source of error by emphasizing the benefit of having correct information for the user and the negative effect of incorrect data. A summary of errors resulting from input forms is shown in Figure 4.

Figure 3

EXAMPLE OF PRE-PRINTED, PRE-PUNCHED CARD


1

Please enter the following information if it is not printed correctly on the top line of this card

3 3 0 - 2 8 - 6 0 9 3

Last name first middle initial

Figure 4

SUMMARY OF ERRORS IN INPUT FORMS

Type of Error	Example	Solution
Poor instructions	Errors in filling out forms	Design forms carefully Test forms carefully Representative sample of users
Unauthorised use	Friend registering for a student	Control distribution and collection of forms
Poor motivation	Person not bothering to read the instructions	Emphasize benefit of correct data and negative effect of wrong information

2.3 Data Collection

One of the most important places for quality control in data-processing systems is at the point of data collection. Omissions and inaccuracies in source-information documents can be reduced by having trained personnel inspect the forms at the point of collection. For example, the arrangements for registration of students can include stations where the information submitted can be checked while the student still is available for correction of errors or omissions.

Not all inaccurate information can be caught by visual inspection, as in the case of two individuals using the same Social Security number. One solution to this specific problem is the use of a "turnaround" document with pre-entered data to preclude errors and omissions. In some systems, a card pre-printed and pre-punched with basic information such as identification number and name may be used; an example is shown in Figure 3. When this card precedes others to which its basic information applies, it is called a "header card".

Another type of inaccuracy involves entering data in the wrong place, such as putting a sex code in the field intended for a marital code name. If a visual check does not catch this error, it can be caught by a validity-checking program and careful design of codes.

Some errors are caused by bad handwriting. Some numbers and combinations of letters and numbers are easily misinterpreted, such as 51, 57, and S1. Boxed spaces on input forms can help to avoid this problem. A common mistake is reading the letter "O" in lieu of the numeral zero. Standardized practices, such as slashing the letters "O" and "Z" to distinguish them from the numerals "0" and "2", can be helpful.

Data can be physically lost after its collection. In one institution, a large number of registration cards disappeared in the process of transferring them from the place of registration to the data-processing building; as they were not found, the data had to be regenerated at great expense and inconvenience. This type of problem can be eliminated or minimised by appropriate logging and checking.

Other errors include incorrect coding and duplication of entries. Such errors can be reduced by proper training of coders, and by establishing and enforcing procedures. All these errors and their solutions are summarised in Figure 5.

Figure 5

SUMMARY OF ERRORS IN DATA COLLECTION

Type of Error	Example	Solution
Missing data	Data omitted from form	Train clerks to check forms
Inaccurate data (data too long and hard to remember)	Social Security number not remembered	Turnaround document Use header cards
	Incorrect transcription of name	Use validity program
Bad writing	Difficult to distinguish between handwritten 51 and S7; OIZ and O12	Standardize symbols Boxes for numbers Type when possible
Lost data	University lost registration forms in transit from one building to another	Log all data Check completeness Bond key employees
Coding errors	Wrong code used Omission by coder	Train the coders Verify the coding Improve code books
Duplication of entries	Forms submitted twice	Set up and enforce control procedures

2.4 Data Preparation

Data preparation consists of converting information from source documents into machine-readable form. Conversion is usually done on a keypunch machine, a key-to-tape device, or a keyboard terminal that communicates with the computer.

There are three important rules to follow to reduce the errors that can occur at this stage:

i) Give careful attention to selection of the personnel for input preparation. They should be motivated and trained on the job for specific applications. The work does not require the highest skills, but it is extremely prone to error.

ii) Design and enforce procedures to identify errors, trace them, log them, correct them and then take steps to prevent these errors recurring. As mistakes certainly will occur, they must be anticipated and traced. The procedures must be flexible to allow for handling unforeseen types of errors.

iii) Set up and maintain liaison between the department that is the source of the data and the department that uses the data. Very often, of course, one department is both the source and the user. If not, then the using department is usually well motivated to improve its liaison to help in minimising errors at the source.

As most data are converted to machine-readable form by keypunching, the first place to check for errors is at this point. It is important to have easily-understood keypunching instructions, including visual aids. Even an excellent operator using the best instructions will make errors, however, so that it is also important to have the work verified by a different operator.

Key verification requires about as much time as did the original keying operation, but the increased accuracy may make the extra time and effort worthwhile. Most keystroke errors can be detected by this method, especially when the verifying operator is one other than the original operator.

A self-checking number feature is available for some of the data-preparation devices. It provides the machine with the validity-checking capability of the kind described as the "check digit" system in Section 2.5.

Another method of data editing is sight verification, or reading. In some applications, particularly those involving textual material, many keying errors can be caught by sight verification performed by clerks. This method is used for verifying the accuracy of data displayed on video terminals.

Other errors in preparing input can come from handling cards, which may be misplaced, disordered, damaged, or duplicated inadvertently. Such problems can be minimised by having sound, well-tested, well-enforced procedures, and by proper training of employees.

In some cases, a reassignment of the responsibility for data preparation can result in reduction of errors. A case in point is that of a system that consistently showed errors in coding and keypunching in about 15 per cent of its input. The input was keypunched in the data-processing centre from forms that had been coded by clerks in the using department with information from transactional records; responsibility for accuracy of the input was divided between the user and the centre. Changing this arrangement to allow the user to have full responsibility for data preparation resulted in reducing the error rate from 15 per cent to two per cent.

The changes in the case described above involved placing keypunch machines and operators in the office of the using department. The coding forms were eliminated, and the input cards were keypunched directly from the source documents. The cards were given in batches to the processing centre for editing by computer, after which they were returned to the user with error listings for correction by the original operator; corrected data was then submitted as input to the system.

The significant factor was the establishment of responsibility for the input data; in the former arrangement, each department tended to blame the other. It should be noted here that not all users have a sufficient volume of work to justify having their own hardware.

Conversion errors are summarised in Figure 6.

Figure 6

SUMMARY OF DATA CONVERSION ERRORS

Type of Error	Example	Solution
Keypunch error	Incorrect data in machine-readable form	Train keypunch operators in job Have good working conditions Provide good procedures manuals Have work verified by a second operator
Handling cards	Dropping cards Misplacing cards Erroneous duplication of cards Loss of cards	Log data Enforce standard procedures
Inaccurate data from terminals	Incorrect data keyed	Sight-check before sending to processor Use validity program
Hardware error	Incorrect data	Proper maintenance

2.5 Programmed Detection of Errors

After converting and verifying the input data, remaining inaccuracies can be detected by programmed checking of the validity of entries and computed results. A summary of such checks is shown in Figure 7. The three principal categories of errors are inaccuracy, unreasonableness, and omission.

Some inaccuracies occur because a simple transposition of digits or mis-keying of digit in entering or transcribing data. Many of these errors can be detected by use of a self-checking arrangement in which a predetermined digit, called a "check digit", accompanies the numeric field. The check digit is calculated by a formula based on weighted positional values; once correctly established, the check digit is carried along to serve as an aid in verification. When the information is being processed, a computer program can perform the identical calculation and compare the result with the input check digit. Check-digit systems will detect up to 95 per cent of the errors caused by transposition and mis-keying of digits.

Other errors occur when characters appear in a field where they are meaningless, such as alphabetic characters in a numeric field, numeric data in an alphabetic field, or special symbols in the wrong field. Most of these errors can also be detected by computer programs.

Figure 7

SUMMARY OF PROGRAMMED DETECTION OF ERRORS

Type of Error	Example	Detection
Inaccuracy	Numbers in name field	Numeric data in alphabetic field
	Number contains alphabetic entry	Alphabetic data in numeric field
	Transposition of characters	Check-digit system
Unreasonableness	Incorrect value	Range check
	High school graduation before birth	Logical check
	Unexpected changes	Historical trends
Incompleteness	Birthdate missing	Check for omission
	Student grade missing	Check master record with transaction

Data can be checked for reasonableness by programming to compare a given value with a predetermined range of acceptable values for that data element. The automatic checking can provide for different actions to be taken according to the result of the comparison: "value is acceptable as is"; "value is possible but out of the ordinary"; or "value is unacceptable".

For example, consider a data element meant to show total number of contact hours taken by a student in any one term. Values from 1 through 20 may be specified as typical; values from 21 through 26 as possible but unlikely; and all other values as definitely unacceptable. The validity-checking program would then allow the processing of data with values from 1 through 26 inclusive as valid, but would identify those records with values of 21 through 26 for further checking by the appropriate person; the program would reject records showing less than 1 or more than 26 credit hours, and produce a list of them to indicate the errors that should be corrected.

Reasonableness of some data values can be checked by use of historical data. For example, consider the departmental account that normally shows monthly expenditures of about $8,000. If the total for the month being processed shows an increase in expenditures of 100 per cent, it may be worthwhile to check the account for errors, or to mark it for special attention. The Internal Revenue Service in the United States uses similar techniques in processing tax returns.

Comparisons may be programmed to determine whether input records have the proper relationship among themselves or with a master file. An example is a programmed routine to check a student's date of high school graduation against his date of birth. These dates are often reversed on applications for admission.

Erroneous omission of data can be detected by programmed checking of fields that should not be blank, or reference to a master file to determine whether transactions are missing. The programmed action to be taken on an incomplete record can depend on the gravity of the omission. For example, an entrance test score may be considered essential in admission processing, and a parent's telephone number may be treated as unimportant; records without the essential information may be rejected, and those lacking only the unimportant data may be processed, with appropriate listings of the items requiring correction.

The computer programs used in checking the validity of data should provide for external specification of the ranges and other criteria so that changes may be accommodated without reprogramming. Careful specification of the terms and conditions of acceptability is important.

2.6 Operations

Controls in the operations phase of data processing must be exercised to prevent occurrences of unauthorised use of the computer, processing at the wrong time, fraud, sabotage, and accidents. Figure 8 summarizes these controls.

Wrongful access is a serious problem in data processing. This includes not only the matter of an unauthorised person gaining access to the computer, but also the matter of a regular employee processing data to which he should not have access. To avoid the former case, access to the system must be carefully controlled; all activities should be supervised and logged for review. As to the latter case, documented control procedures should be in effect; this will be discussed later.

Another problem is that of processing being incomplete because of data having been delayed, lost, or destroyed. To detect errors of this kind in an updating situation, various totals from the old file, the transactions, and the new file must be checked. The documented procedures should call for logging and checking the applicable counts and totals, and should include instructions for the action to be taken when errors are found.

Although it is impossible to eliminate all opportunities for fraud or malicious mischief, the chances can be reduced by precautionary measures. Some suggested controls are:

a) Whenever data processing is going on, there should be at least two employees present. This rule should apply not only as a security measure, but also as a matter of good safety practice.

b) Working schedules, of either the operators or the jobs to be processed, should be arranged so that no one operator handles the same programs over long periods.

c) As far as possible, avoid assigning an employee to work with the kind of data that could lead to a conflict-of-interest problem. For instance, a student should not be given a job in which he could illicitly change grades.

d) All actions taken by an operator that affect the logic of processing or that alter financial records or other vital data should be recorded on equipment that is not accessible to the operator.

e) Sensitive files, such as payroll and student records, should be accessible only after proof of authorisation, and then only on a sign-in-sign-out basis. Some systems for computers provide a file-protection feature to deny access unless the proper password is given.

f) Programmers and systems analysts should not be the "production operators" who process data for customers of the computer centre. After a computerised system has been properly developed and installed, its programmers and analysts should be divorced from its routine operation.

g) Team programming should be encouraged for those programs that involve sensitive information.

h) The program used in processing should be compared from time to time with a library copy of the program that is kept by

Figure 8

SUMMARY OF OPERATIONS CONTROL

Type of Error	Example	Solution
Unauthorised access	Unauthorised person having access Authorised person having access to unauthorised data	Control access Log all activities Rotate work-shift assignments Have at least two people on duty
Data not processed when it should be	Documents left on clerk's desk or misrouted	Test: Old record-count plus additions minus deletions must equal new count
Fraud	Changing grades Payroll cheating	Control access to files and programs Auditing Removing conflicting interests Vary work schedules Strict supervision Bond personnel
Sabotage	Processing centre burned by rioters	Heat & smoke alarms Fire extinguishers Panic switches Intrusion detectors Police patrol Training in emergency procedures Have at least two people on duty Safeguard all doors and windows; steel or steel mesh
Acts of nature	Flood Storm	Back-up equipment Shut down or protect with special devices

the development unit or control group to see if any unauthorised changes have been made.

i) Periodic audits should be instituted.

The suggested controls will aid in detection and prevention of the problems of fraud and malicious mischief. Knowledge that precautionary measures are being taken is often enough to discourage any attempt at tampering.

Protection from sabotage merits serious attention. Many institutions of higher education have had dangerous attempts at deliberate damage to data files or equipment during times of campus unrest. To guard vital files against sabotage, as well as natural disaster, it is recommended that each important, irreplaceable file have a duplicate stored in a secure vault, preferably at an off-campus location. As to physical protection of the data-processing centre, some items to be considered are: silent "panic buttons" to signal for police help; intrusion detectors to set off an alarm in the police station, or police patrols, while the centre is closed; fire and smoke detectors; fire extinguishers; fire-resistant construction; entry control; key and lock control; training in emergency procedures; steel-mesh covers for windows; steel doors; and special identification systems(1).

In the case of a catastrophe such as flood, earthquake, fire, or explosion, emergency procedures should be put into effect immediately. Employees must be trained in these procedures, and the procedures should be reviewed regularly. From the standpoint of recovering to enable vital processing to take place, the only safeguard is to have complete and well-tested back-up facilities arranged with another centre.

2.7 Data Files

As previously mentioned, vital files must be kept secure with duplicates stored in a different location. This procedure will provide back-up in case of sabotage or disaster(2).

There are other factors concerning the security of data. These are summarised in Figure 9.

One of the important items to be considered is the physical condition of files. If the file consists of punched cards, it must be stored in a clean place with the proper humidity, and the cards must be kept under proper pressure to avoid warping. If the file consists of data recorded on magnetic tape or disk, the magnetic medium must be kept under "clean-room" conditions subject to rigid control of temperature, humidity, and cleanliness.

Responsibility for the condition and the content of data files should be clearly assigned. Centralising the control of files with a librarian offers the advantage of clearly defining the responsibility for enforcing set procedures for storage of and access to the files.

2.8 Programming

Many errors in computerised systems can be attributed to faulty programming. Such errors are often difficult to discover and can go

---

1) For the approach taken by Fresno State College in the United States after its computer centre was destroyed by students, see Computerworld, 8th July, 1970, p.8.

2) For a case of magnetic-tape sabotage, see Keller, A.E., (1970), "Aesop's Fable", Business Automation, January, p.112.

Figure 9

SUMMARY OF DATA FILE CONTROLS

Type of Error	Example	Solution
Physical condition of records	Warped cards	Controlled-humidity storage
	Dirty tapes or disks	Special cabinets Periodic cleaning
Acts of nature	Fire Flood	Special vaults Back-up data
Theft, fraud, or sabotage	Changing a master record Rioters destroying vital files Illicit access to sensitive data	Control access to all data files: Data librarian Lockwords Control labels Back-up files Destroy material no longer useful

undetected for quite a long time. As Davis (1969: 428) said, "Large programs are sufficiently complex to harbour latent errors which may not show up for weeks, months, or even years after the program has been accepted".

One of the most common programming errors is the use of a faulty algorithm, i.e. a faulty set of instructions for the computer. One way of checking for this error is by comparing test results computed by the algorithm to those that have been computed by hand or by desk calculator. Section 2.5 describes other checking methods such as the comparison of computed results against limits of reasonableness, and the verification of counts and totals by various means.

Another common source of error is the improper use of a program because of poor documentation. This kind of error can be controlled by having and enforcing standards for program documentation. Every program should be documented with a full description and with clear instructions for its use.

Mischievous or disgruntled programmers have been known to make deliberate errors in programs. This sort of problem may be prevented by close supervision or by the use of "team programming", i.e. programming by a group of employees. A periodic review by an auditing programmer would detect and discourage such damage.

Programmed fraud can be classified quantitatively as the "nibble" type (just a little fraud at a time) or the "bits" type (one large fraud). Examples of nibbling fraud are the changing of a few grades or the stealing of small amounts of money periodically. One type of nibbling embezzlement has been referred to as the "salami" payroll fraud, where the dishonest programmer coded a program to use remainders from rounded calculations to increase the amount of his own paycheck. In documented cases, this kind of fraud amounted to less than a cent for each payroll item, but the accumulation provided a handsome bonus for the errant programmer.

Bite-size fraud is exemplified by the embezzlement of a large sum of money or the alteration of a large number of student grades. This sort of fraud may be detected by auditing and checking for unreasonable values or control totals.

However, no matter how well a system is protected, someone will invent a new means of fraud which may be difficult to detect. Constant vigilance is required.

Serious problems in processing can be caused by mistakes that can occur easily in the handling of programs. One out-of-sequence program statement can produce disastrous results in the output. These problems can be overcome by training, care, and strictly following standard procedures in programming. For example, all program decks should contain proper sequence numbers, even in early test phases; the probability of a deck getting out of order is low, but the cost of the resulting error can be very high. Finished programs should be kept in magnetic form, if possible, rather than in cards. A summary of programming errors and their solution is shown in Figure 10.

Figure 10

SUMMARY OF PROGRAMMING ERRORS

Type of Error	Example	Solution
Wrong algorithms or wrong programming instructions	Program produces wrong answers	Testing: Desk check Manual check Compare historical results Cross-check totals
Problems caused by compiler	Peculiarities of programming language give wrong results	Careful testing Program in two different languages
Poor documentation	Maintenance work not documented	Set standards for documentation Enforce standards
Theft or fraud (nibble or bite)	Unauthorised changes in programs for grades and payroll	Check program against original version periodically Team programming Auditing
Bad processing	Wrong sequence of cards	Training and care on processing Procedures to be set up and enforced

2.9 Processing

As most processing is performed by operations personnel, processing control is closely related to operations control. There are, however, additional controls not related to operations which can be designed into systems. Some of these controls are summarised in Figure 11 and are self-explanatory.

Figure 11

SUMMARY OF PROCESSING CONTROLS

Type of Error	Example	Solution
Processing errors	Records lost	Programmed checks: Record counts Transaction counts Train personnel
Outdated file	Using old master file	Generation data sets (keep track of files automatically) Use standard labels for all files
Shortage of supplies	Lack of grade-report forms at time of printing	Proper planning and inventory control

2.10 Output

Output is the product of all input and processing. If proper control is exercised in each of the steps discussed earlier, then output should be free of error. Nonetheless, all output should be carefully checked, and proper controls should be built into each system. Some of these controls are listed in Figure 12.

A cursory sight-check for obvious mistakes in output may save much embarrassment. At one institution, a last-minute, untested change in a computer program caused all of over 1,400 student grade reports to be addressed to and delivered to one person. In a similar case, the mailing labels for the alumni newspaper were correct as to name and street address, but all were addressed to and delivered to the same small town. Embarrassing and expensive situations such as these can be avoided by seeing that output is checked by alert operators.

Another method of controlling output is by checking for completeness. This can be done by verifying the number of pages of output, checking control totals for each process or report, or by following appropriate end-of-report messages.

Many of the programming techniques discussed earlier under the checking of input can also be used for controlling output. The checks

Figure 12

SUMMARY OF OUTPUT CONTROLS

Type of Error	Example	Solution
Obvious errors	All mailing labels bear same address Incorrect data or totals	Sight checks Defer large-volume printing until proof data checked
Incomplete results	Items or report pages missing	Check totals and page counts
Inaccurate results	Total of count by student class not same as total count of students Employee claims sick pay but time card shows full day of work	Program to check for completeness, reasonableness, consistency, and accuracy Interfile comparison Sample check of output with corresponding input

for completeness, reasonableness, and accuracy used for input control can be used similarly for output control.

Output can also be checked for logical inconsistencies. For example, a payroll record that shows sick leave for an employee on the same date that his time card had a work entry is obviously in error. For another example, course records indicating enrollment of the same student in two different sections of the same course in the same term are very likely in error.

Other checks for consistency can be programmed to check control totals or crossfootings. For instance, if the total of the count of students by class does not equal the count of all students on a report, an error exists in either the file maintenance program or the report program.

2.11 Administrative Responsibility

Although some of the control features discussed in this paper are of a technical nature, it is very important for higher education administrators to appreciate the capabilities and limitations of such controls so that they may be used appropriately. Failure to do so will result in the loss of the reliability and credibility of the system.

The administrator should be an active participant in the matter of quality control. He should help identify points of control, assign persons to be responsible at the points, and ensure that the control procedures and standards are enforced. He should also see that procedures are designed to minimise the chance of error, to detect errors promptly when they do occur, to halt their proliferation, to correct them, and to prevent their recurrence.

One important control principle is that no one person is to be allowed to perform all of the functions in processing vital transactions such as payrolls or student grades. In the case of payroll, no one person should record the number of hours worked, prepare the input data, write the payroll program, operate the computer, and distribute the checks. More than one person or administrative unit must be involved with proper control at each step. This rule was well stated by Porter (1965: 55-63): "One of the basic principles of internal control is the separation of the people who authorise a transaction, those who have custody of the assets required, and those who record the accountability for the asset".

Applying this principle to the organization of information systems, one should arrange duties that would provide for separation of the five basic functions: programming and systems development; handling of input media; operation of data processing equipment; documentation and file library; and handling and distribution of output. Such separation may not be possible in small institutions but it should be done whenever possible. Good organization is fundamental to good control.

Another problem that the administrator must face is that of security. He must decide which information must be protected, the extent of the protection, and how the protection is to be accomplished. He must determine who is to be given access to which data for what purpose. He must see that the standard procedures for security are followed so that violations will be detected, stated penalties applied, and the proper corrective action taken.

In making the rules for the security of an information system, it must be realised that there is a trade-off: the greater the protection, the greater the cost in time, money, difficulty, and annoyance to the user. The rules regarding security should not be so lax as to make a mockery of it, nor should they be so strict that proper access is prevented or unreasonably delayed. The rules should be flexible, selective, effective, and enforceable.

Quality control is a continuing responsibility of the administrator and the data processing manager. They must periodically review the existing controls for adequacy. The review should include an analysis of the occurrence of problems, and ask questions such as: How often did it occur? What was the cost of the error? How and why did it happen? Could it have been prevented? If so, how? What will better procedures cost? Depending on the answers to these questions, the controls may be either tightened or relaxed.

3. SUMMARY AND CONCLUSIONS

The quality control functions necessary at each stage of processing are discussed in this chapter. The controls examined are those for documentation, forms, data collection, data preparation, operations, data files, programming, processing, and output. The many types of errors which can occur at each point are discussed, and controls suggested to minimise or eliminate the errors. These points are summarised in tables following the sections.

In considering the numerous techniques that have been discussed, one can easily see that the difficult problem is not how to control, but how much to control. Too much control is costly; it can impede work and affect morale. Too little control reduces the usefulness of a system because of inaccuracies. The trade-off must be determined, and the decision made accordingly.

Quality control of information is extremely important for every information system. It must be planned and implemented as part of the original development of the system. It must also be monitored and maintained throughout the life of the system.

To control the quality of information, the subject of this chapter; to evaluate and maintain an information system, the subject of the previous chapter; and to ensure the orderly operation of the system, it is necessary that an information system be organized. This requires not only an organizational structure for the information system but a desired relationship with the organizational structure of the institution. These structures, and relationships, are the subject of the next chapter.

REFERENCES

Davis, G.B., (1969) Computer Data Processing, New York: McGraw-Hill Book Company, Inc.

Eckman, T. and C. Froberg, (1965) Introduction to ALGOL Programming, Lund, Sweden: Student Literature.

Porter, W.T.Jr., (1965) "A Control Framework for Electronic Systems", The Journal of Accountancy, October, pp.56-63.

Chapter 9

ORGANIZING INFORMATION SYSTEMS

by

K.M. Hussain, New Mexico
State University, U.S.A.
and
C.R. Thomas, College and
University Systems Exchange,
Colorado, U.S.A.

CONTENTS

1. **Introduction** 172
2. **Organizational Roles in System Design and Use** 172
3. **Interrelated Systems** 174
4. **Data Processing Organization** 176
5. **Organizational Attitudes** 181
6. **Organization of the DED/DD System** 181
    6.1 **Control of the DED** 182
    6.2 **Data Administration** 183
    6.3 **Resources for Data Administration** 187
    6.4 **Policy Analyst** 188
7. **Summary and Conclusions** 190

References 191

Annotated Bibliography 323

1. INTRODUCTION

The acquisition of a large scale computer and the establishment of an administrative systems office will not automatically guarantee that an institution of higher education will reap the benefits of information systems technology. While adequate computer hardware and a competent systems department are certainly necessary, they are by no means the only ingredients required for the development of successful information systems.

Organizations, their roles, structures and attitudes are probably the most important consideration in the successful employment of information systems technology in higher education. This chapter will discuss several of the major concepts relevant to organizing information systems.

2. ORGANIZATIONAL ROLES IN SYSTEM DESIGN AND USE

It is only recently that the word management has become acceptable in the field of higher education. Even today a preferable term is administration. This subtle semantic difference seems to imply that higher education administrators are significantly different from their commercial counterparts. This traditional collegial environment must be understood if the implementation of information systems in higher education is to be successful.

The major roles in the development of information systems are played by the system user and the data processing specialist. Leon Albrecht states: "The responsibility for developing a systems design plan should rest with the head manager of the information systems division(1). This statement is typical of the commercial approach to system development. Practical experience and observation indicates that in higher education the system user must be intimately involved at the systems design stage. The very finest system design will encounter operational difficulties if it is forced upon the using department.

There is, of course, a price to be paid for the involvement of the system user at the design stage. Information systems technicians must communicate in language understandable to the user rather than in technical terms. In many cases, time must be spent educating

---

1) Albrecht, 1973, p.183.

the user in technical concepts. This price is often frustrating to the technicians, who wish to charge ahead with what they view as the solution to the user's problem. In a commercial environment the alternatives available to the user usually are to accept the technician's solution or make room for someone who will accept the solution.

In higher education the information systems developers are generally placed in a service role and must work with the user to solve his problem. Failure to operate in a co-operative mode can become expensive. Generally the technicians provide successive solutions for a problem until they hit upon the solution the user had in mind, or the user accepts a solution that does not meet his needs. Institutional needs in the area of information systems are best served by co-operative efforts between the system user and the data processing specialist. There is, however, need for co-ordination between the major systems. This role is generally assigned to a management systems office, an office of administrative studies or some other organizational unit reporting to a second level chief administrator. This office is charged with co-ordinating the development of operating information systems so that they both serve the needs of the user department and provide information for institutional analysis and planning. Usually this office is also responsible for the maintenance and production of an institutional data element dictionary and a uniform code manual. The management systems office generally works with the other operating units of the institutional administration in a persuasive mode; however, occasionally it must be in a position to exert some control over system design when it is in the best interest of the institution to do so. Such control will generally be exercised through a second level chief administrator rather than directly over the operating department.

Additional organizational considerations occur in the information systems production processes. Data and files are considered to be "owned" by the major administrative units and are only stored and processed by the administrative data processing unit. To put it another way, the administrative data processing department assists each operating administrator in accomplishing his mission; it does not accomplish it for him. As stated by Caffrey and Mosmann, "It is of utmost importance for the president to understand that the fact that automated processes are used, does not relieve the administrative staff members of any of their primary responsibility for the operational efficiency and effectiveness of the functions they perform. It is still they, not the computer centre or its staff, who are responsible for running the college and its programs"(1).

---

1) Caffrey and Mosmann, 1967, p.183.

Unfortunately many line administrators abdicate their responsibility for decisions over the functions of their office the moment their procedures are placed upon the computer. It is the nature of computer personnel that, given the opportunity to make decisions for other people, they readily do so. In such cases the technician then effectively becomes the line administrator.

If organizational roles are properly defined, the user department maintains control over data origination, production schedules and output distribution.

In the case of an administrative department that distributes reports to units throughout the institution, that administrative department should remain in the position of answering all questions about the computer output. Frequently the natural reaction of recipients of reports is to call the computer centre with any questions concerning a computer printed document. When this occurs, two mistakes are made. First, the computer department personnel may provide a wrong answer; and second, a new and erroneous information flow path is established.

There are very simple techniques that can be employed to avoid this type of problem; for example, printing the telephone number of the appropriate administrative department on each page of a computer-produced report. Then, of course, the administrative department must answer questions rather than refer calls to the computer centre, or no advantage has been gained.

3. INTERRELATED SYSTEMS

As mentioned earlier, there must be one office responsible for the co-ordination of the development of operational data systems. It is important that this office report to an administrative level that is at least equal to the reporting level of all of the major operational data system users. It is extremely important that the management systems office do not report to any one of the major users. There are an abundance of examples in the United States and in Europe to indicate that when the management systems function reports to any one of the major operational system users, then that system soon becomes overdeveloped at the expense of other operational data systems in the institution.

While the management systems office should report to a high level in the administration, it is still important that it operate in a co-ordinating mode, rather than dictating systems solutions to the operational users. One organizational technique for providing co-ordination between the operational data systems users is the formation of a committee of the major administrative departments.

In the United States these committees are called by various names, such as Administrative Systems Council, MIS Council, Committee on Institutional Data, Administrative Data Committee, and other similar titles. The structure and role of such a group is important if an institution is to achieve the appropriate interrelating of operational data systems to support management decisions.

For the purpose of illustration we shall call this group the Administrative Systems Council and first discuss the membership of the group. Of course, the heads of all of the major administrative systems users should be members, including the management systems office. In addition, representatives of each of the levels of management of the institution should be part of the group. All schools or colleges and departments will utilize information from the operational data systems. These offices are sometimes referred to as the "user's users" since they receive reports from the fiscal office, the registrar's office, and other administrative offices. Normally it is not required that representatives of all operational units at the "user's user" level serve on the committee. Perhaps this duty can be rotated among the heads of schools and colleges and heads of departments. The important point is to ensure that all levels of management are represented on the Administrative Systems Council.

It should be clear from the outset that the role of the Administrative Systems Council is advisory concerning system development priorities and inter-system co-ordination. The committee should be chaired by an administrator one organizational level above all of the members of the committee. It also must be made clear that the committee is not constituted to provide for day-to-day supervision of computer activities.

The Administrative Systems Council should meet on a regular basis to review progress of systems under development and to advise the administration on the prioritisation of new systems development needs. Proposals for the development of new administrative systems should be reviewed by this group. In addition to providing a more co-ordinated approach to system development, some spin-off benefits will accrue to the institution. The various administrative organizational units of the institution will acquire a better knowledge of the operation of other units and of the total administrative process. In addition, system designers and the user department staff will achieve a more professional status through the presentation of systems proposals to the Administrative Systems Council.

It is very important, however, that the Administrative Systems Council should not be treated as, or operate as, a "rubber stamp" committee. The members and the committee must be viewed as a very important ingredient in the development process for information systems.

4. DATA PROCESSING ORGANIZATION

The first decision most institutions face relative to the organization of computer resources is centralisation or decentralisation of the major functions of academic computer support and administrative computer support. Controversy on this point continues in the United States and in Europe with staunch advocates for both positions. Successful examples of both centralised and decentralised installations can be cited. In his recent book, Charles Mosmann said:

> "Because of the management costs of centralization and the technical costs of generality, many medium-to-large institutions with small research budgets accept some degree of decentralization. Separate facilities will be available for academic and administrative use; some mini-computers will be in use in various science departments; some users will occasionally buy remote computing from a commercial or regional center. The real function of the central organization is to foster good computing and not just to create an orderly situation. If apparent confusion and duplication promotes good research and good education, then multiple centers should obviously be encouraged. Yet by and large this is not often the case with multiple centers and competing services. Attempts to create order are generally attempts to improve the quality of computing and the quality of the research and education it serves. At smaller institutions, of course, centralization may be a necessity. Financial resources may be simply insufficient to permit the dispersion of computing, while individuals or small groups of users may not have the independence to be able to override the interests of the majority. Thus at these institutions the job of co-ordinating computing is considerably easier, since almost all the users will agree that co-operation is to their own best interest."(1)

There is a perceptible trend in the United States towards the computer utility concept, where the operation of the computer hardware and its associated systems provide computing resources to all users. Separate organization units make this computing resource available to the academic and administrative users. Figure 1 shows a typical organization as presented by Mosmann(2). In this hypothetical organization the director has six separate organizational units reporting to him. A more detailed description of this organizational chart is available in Mosmann's book.

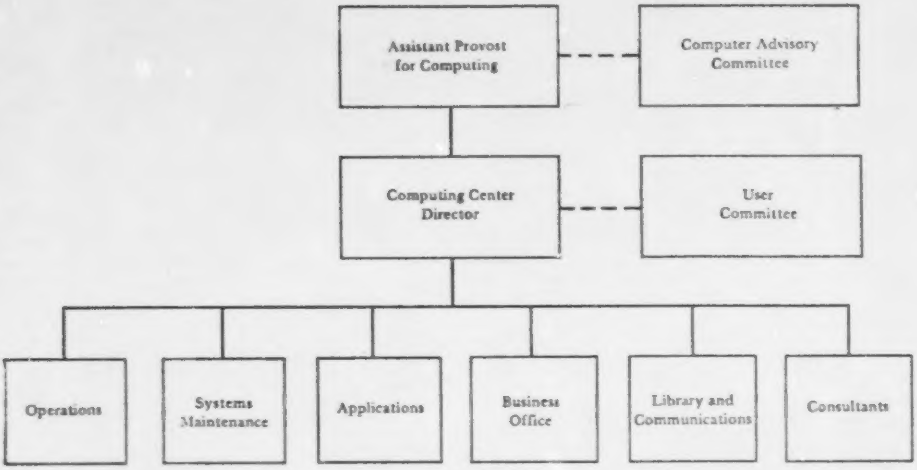
Figure 2 provides another view of a typical computer centre organization and Figure 3 shows further details for the system's

---

1) Mosmann, 1973, p.66.

2) Mosmann, 1973, p.71.

Figure 1


Assistant Provost for Computing
Computer Advisory Committee
Computing Center Director
User Committee
Operations
Systems Maintenance
Applications
Business Office
Library and Communications
Consultants

development division. These charts were developed by Dr. Robert E. Taylor, Associate Director of the Computing Center at Virginia Polytechnic Institute and State University(1).

The new and emerging role is that of a data administrator shown in Figure 3. This role and the role of another emerging position - the policy analyst - will be discussed later under administration of data.

Whatever conceptual model is chosen for the organization of data processing there are four major functions to be performed:

1. Management
2. Systems Development and Programming
3. Operations
4. Production

The management of the data processing organization (or information service), as stated earlier, must report at a high enough level in the organization to accomplish co-ordinated development at the institutional level between various administrative information systems. While the director of the data processing organization must have the political capability to deal with other organizational units, he must also have adequate technical knowledge to ensure that tech-

---

1) Taylor, 1973, pp.39-70.

Figure 2

A TYPICAL COMPUTING CENTER ORGANISATION CHART

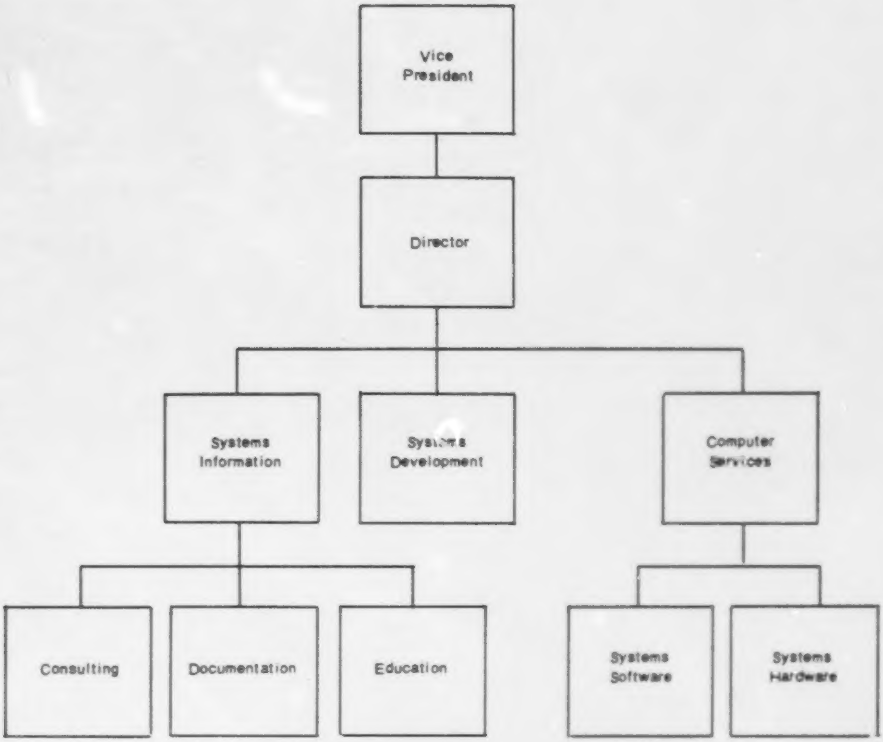
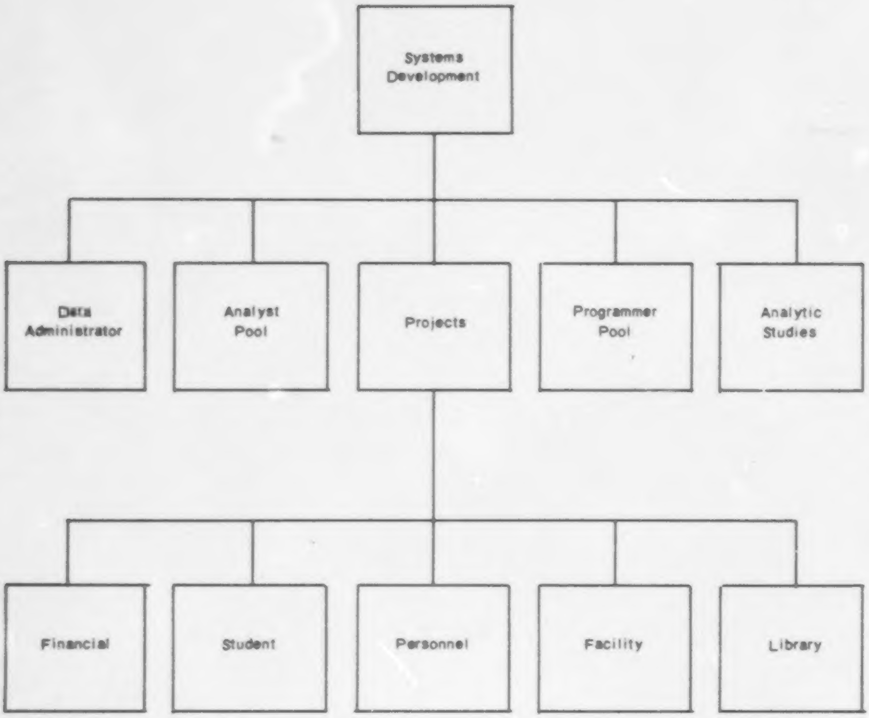

Vice President
Director
Systems Information
Systems Development
Computer Services
Consulting
Documentation
Education
Systems Software
Systems Hardware

Figure 3

SYSTEMS DEVELOPMENT DIVISION


Systems Development
Data Administrator
Analyst Pool
Projects
Programmer Pool
Analytic Studies
Financial
Student
Personnel
Facility
Library

nical considerations do not override the system development and production process. Unfortunately, many superior technicians are promoted to management positions, causing the institution to lose twice. First, they lose the services of a good technician, and second, they may acquire a poor manager. When faced with the task of hiring a new director for the data processing organization, institutional administrators would be well advised to obtain the services of other successful directors as consultants to assist at the interview stage.

The systems development and programming division of a data processing organization is normally divided along the functional lines of a financial systems unit, a student related systems unit and additional units to handle other major systems. These units work with the user in the system definition and design stages then follow on with the detailed system design, programming, documentation and implementation steps. The end-product of the development effort is a fully documented, tested and operational system turned over to a production section.

The operations unit of a data processing organization includes the computer hardware operation and systems programming; that is, the generalised software associated with computer scheduling, compiling and other tasks necessary to provide computer resources to each of the users. It should be noted that there is a considerable difference between the programmers who work on operating systems software and the programmers who work on application systems for using departments. Normally system programmers are highly skilled technicians who deal with the very detailed systems software, while applications programmers deal with much higher level languages and are more concerned with the user's viewpoint than with the technical process internal to the computer.

The production function in administrative data processing organizations is sometimes designed as a part of the operations department and other times as a part of the user department. It may also be a separate unit of the data processing organization. The function, however, is to prepare the input for a particular administrative system, receive the output and verify its accuracy, ensure that the appropriate files were used, and other activities related to quality control. The organizational placement of this function is not as important as the fact that it be performed by some unit. It is this unit that is charged with avoiding those disasters everyone hears about. For example, the scheduling of all students into a single classroom at a single hour, or the production of a grade report with all failing grades for all students.

Many models for organizational structures exist; however, the organizational structure for any institution is not as important as the people who fill it. The best structure will not work for incompetent employees and any structure will work for those who have competence.

5. ORGANIZATIONAL ATTITUDES

Successful information systems operations have in common some organizational attitudes that are worth examining. Each of the user organizations, the admissions unit, the student records unit, or the accounting unit, should view their operation as a supporting unit to the institutional management team. If some organizational unit of the institution requires that additional data be collected in the admissions system then the admissions officer's attitude must be "if someone else requires this data, and the resources can be justified and made available, then my unit will be responsible for the collection and maintenance process". In many cases operational managers in institutions of higher education are concerned only with collecting and maintaining the data necessary to perform what they view as their function. The formation of an administrative systems council, discussed earlier, can contribute to improved attitudes on the part of the operational managers in an institution.

It is important, also, that the data processing organization operate in a service mode. The allocation of the system development and computer production resources must be consistent with institutional priorities. As mentioned earlier, the best place for determining these institutional priorities is normally through an administrative systems council where representatives of all user groups can be heard. It is worth noting that resource allocation is not a problem if resources are unlimited. Procedures for resource allocation should exist, however, before unexpected limitations force an unorganized approach to prioritisation of activities.

6. ORGANIZATION OF THE DED/DD SYSTEM

A DED/DD (A Data Element Dictionary/Data Directory) was discussed in an earlier chapter. Developing a DED/DD system of the dimensions discussed requires considerable organization. It raises many questions such as: who should control the content of the DED? What procedures and mechanisms are needed for such control? What equipment and software are required for developing and maintaining the DED and the directories? What personnel and qualifications are required for using the equipment and software? Should these personnel report to the data processing manager or to non-data processing management? These and other organizational problems are the subject of the remaining part of this chapter.

6.1 Control of the DED

One mechanism for the control of the DED is shown in Figure 4. According to this mechanism a committee decides whether or not each data element should be part of the common data base. The committee also periodically reviews the existing DED for redundancies and duplications, and must be responsive to the need for changes in the DED resulting from changes in organisational needs, organisational structure and personnel.

In Figure 4 only one box is used to represent approval authorisation for changes to the DED and the diagram shows the committee as responsible for change approval. In reality, however, the process for approving changes is a sequence of steps involving not only the committee but users, programmers, analysts, and data processing management. The actual personnel involved in approval updating will vary between institutions and even within an institution over time, depending on the organisation's history, the system's complexity and the personalities involved. One approval process for updating the DED, proposed by Cahill(1) is shown in Figure 5.

In controlling the content of the DED the committee has the responsibility of resolving differences between organisational units which arise. In order to perform this duty the committee should be a consortium of users composed of high level administrators so that the committee has the operational power and respectability to enforce decisions made: that is, power to overrule unilateral actions by users, to adjudicate conflicts and to anticipate and overcome resistance.

The committee must also state policy and provide guidelines for the creation, maintenance and monitoring of the DED/DD system. These responsibilities are complex and often very technical. In practice the committee often delegates these responsibilities to a person (or office), retaining merely a supervisory role. The delegated responsibility goes to a data administrator, data manager or data controller, a position which has recently emerged in many organisational structures. The newness of this position is reflected in the fact that a recent survey of the MIS Staffing does not even list this position(2). In 1971 a business firm advertising in the New York Times for a data administrator received, in addition to applications, inquiries from data processors as to the functions and job description of the advertised position.

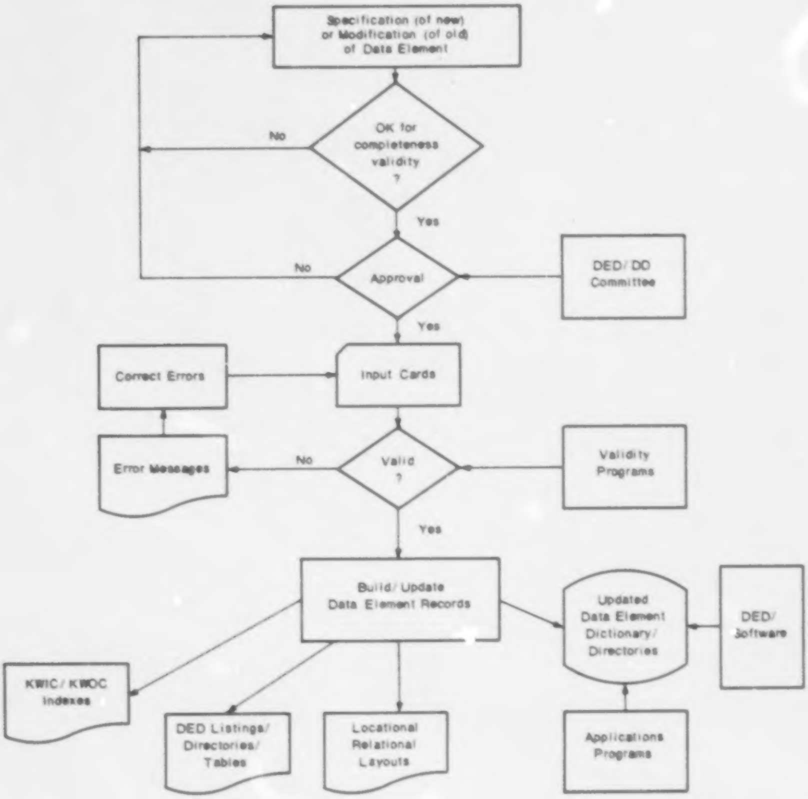
---

1) Cahill, 1970, p.29.

2) Willoughby, 1972.

Figure 4

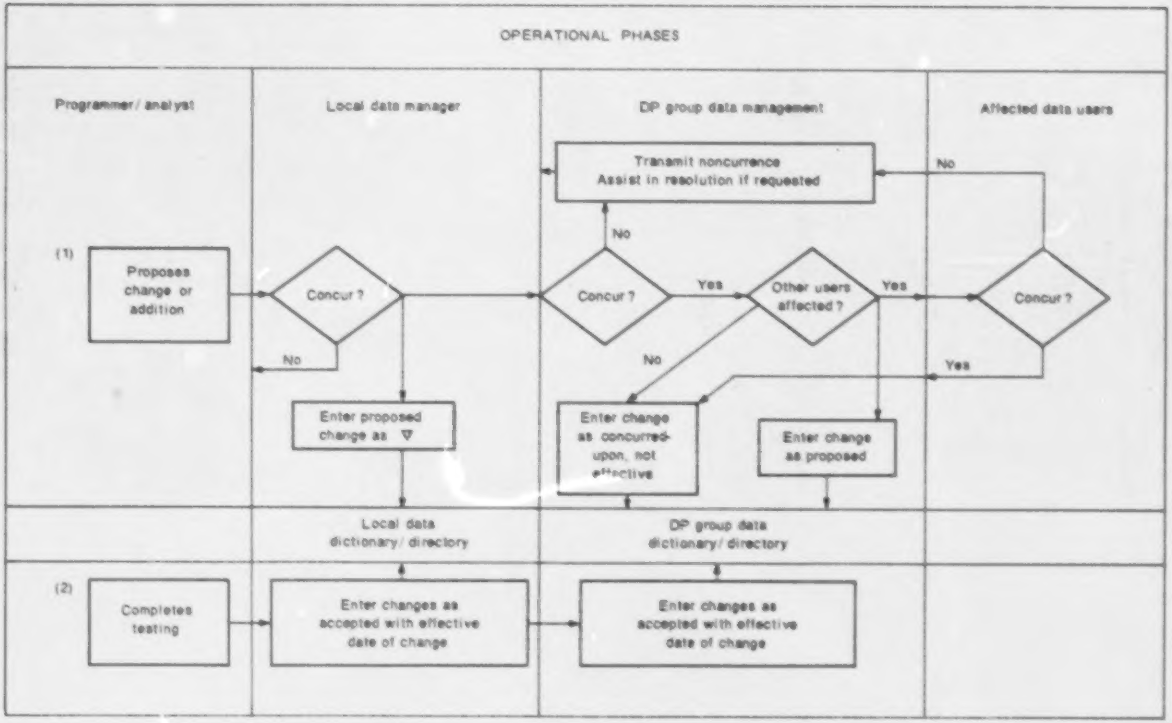
MECHANISM FOR CONTROLLING THE DED



Specification (of new) or Modification (of old) of Data Element
OK for completeness validity ?
No
Yes
Approval
No
DED/DD Committee
Yes
Correct Errors
Input Cards
Error Messages
No
Valid ?
Validity Programs
Yes
Build/Update Data Element Records
Updated Data Element Dictionary/ Directories
DED/ Software
KWIC/KWOC Indexes
DED Listings/ Directories/ Tables
Locational Relational Layouts
Applications Programs

Figure 5

PROCESS OF UPDATING THE DED


OPERATIONAL PHASES
Programmer/analyst
Local data manager
DP group data management
Affected data users
Transmit noncurrence
Assist in resolution if requested
No
No
(1)
Proposes change or addition
Concur ?
Concur ?
Yes
Other users affected ?
Yes
Concur ?
No
No
Yes
Enter proposed change as ∇
Enter change as concurred-upon, not effective
Enter change as proposed
Local data dictionary/directory
DP group data dictionary/directory
(2)
Completes testing
Enter changes as accepted with effective date of change
Enter changes as accepted with effective date of change

6.2 Data administration

All common data bases - even for strictly operational needs - have a need for co-ordination and administration. These needs, however, increase with planning models, especially the needs for maintenance, data security and data control. These needs are being met by an emerging new position, that of the Data Administrator. His role and his functions will now be discussed.

Data maintenance is a very important function of a data administrator and data administration. It requires the constant upkeep of active data, the storing of historical data and the purging of useless data. Data must often be moved between devices and storage media and must be carefully logged and tagged for fast and effective access.

Data maintenance includes adjusting to changes in the Data Element Dictionary and Data Directories.

Data maintenance also includes having the data complete and continuous. The continuity is very important for planning data especially for longitudinal integration. This requires making changes to historical files when definitions of data elements or classifications change. For example, if a department of education were divided into two departments of Physical Education and Education then this change must be reflected in historical data if this data were needed on a longitudinal basis. All changes in classifications and coding must be controlled. Integrity of data is especially crucial in on-line systems. There, a break in processing could result in loss of data. Procedures for back-up and recovery and restart must be established in all such events of breakdown including natural disasters.

Related to integrity is the problem of accuracy and reliability of data. The data base must be monitored for the desired level of quality. Logging and auditing procedures must be established and followed. Performance (on access, utilisation, cost, response time) must also be monitored.

Another important function in data administration is that related to the privacy (misuse of data) and security (unauthorised use of data). The security is important especially with planning data since it is sensitive by nature, has many creators and users. The authorised ones must be distinguished from those unauthorised.

Security can be achieved through passwords, passkeys or other methods of uniquely defining authorised users. The data to be accessed, the time of access, the place of access (such as a terminal) can also be specified. Policies and procedures governing security must be described and enforced. This is a responsibility that is important and growing.

The functions of the data administrator as discussed above and a few other functions are summarised in Figure 6.

Figure 6

FUNCTIONS OF A DATA BASE ADMINISTRATOR

Data Base Design

- Content
  - creation
  - reconciling differences
- Dictionary/Directory
  - create
  - maintain
- Data Compression
- Data Classification/Coding
- Data Integrity
  - backup
  - restart/recovery

Data Base Operation

- DED Custodian/Authority
  - maintain
    - add
    - purge
- Data Base Maintenance
  - integrity
    - detect losses
    - repair losses
  - recovery
  - access for testing
  - dumping
- Software for DED/DD
  - utility programs

    tables/indexes, etc. for end user
- Storage
  - physical record structure
  - logical-physical mapping
  - physical storage device assignments
- Security/access
  - assign passwords
  - assign lock/key
  - modifying passwords/keys
  - logging
  - cryptography
  - modification
- Retrieval
  - search strategies
- Statistics
  - access
  - frequency of processing
  - space use
  - user utilisation
  - response time
- Design operational procedures
  - access to Data Base
  - access for testing
  - interfaces
  - testing system

Monitoring

- Quality of data
  - validity
- Performance
- Efficiency
- Cost
- Use/Utilisation
- Security/Privacy
- Audit
- Compliance
  - standards
  - procedures

Other Functions

- Liaison/Communication with
  - end users
  - analysts/programmers
- Training on Data Base
- Consultant on File Design
- Design Operational Procedures
  - access to Data Base
  - access for testing
  - interfaces

To perform these functions the data administrator needs powers akin to what Lyon calls a "government of the data base with both legislative and executive powers"(1). The legislative role would include power to make rules regarding content definition, access control, security, recovery, logging, dumping and auditing of data. The executive action includes the monitoring and enforcing of rules especially as it concerns security and privacy of data, as well as efficiency and effectiveness of the data base.

To perform his functions the data administrator must have authority over the data base as it relates not only to users in all organisational units but also users at all levels of the organisational hierarchy. This authority under the direction of a supervisory committee extends even over vice-presidents and data processing management, since even they have been found to break the system for personal financial gain. This was found in a study by the Stanford Research Institute. A table from this study is shown in Figure 7(2).

To govern the data base the data administrator needs many talents. The data administrator of the OASIS System at Stanford recommends that a data administrator be a "...technician, politician, consultant ... who speaks with equal fluency the languages of both computers and human beings"(3).

It is difficult to find all of these qualities in any one person. However, data administrators generally have assistance in managing large data bases. They can select a balanced staff to include all these qualities among the staff personnel. In addition, the staff personnel should have expertise as data specialists, as trainers/educators and in public relations and liaison work.

The data administrator and his staff are usually data processing technical personnel. Some organisational theorists recommend they report to a data processing manager. Other theorists, however, argue that the data administrator should report to his supervisory committee (which could be a Users' Committee). This would enable the committee to better control related data processing functions and at the same time give the user the feeling of being the "custodian" of the data base.

6.3 Resources for Data Administration

To administer the data base both special equipment and special software are required. The equipment may consist of coding and decoding (cryptographic) equipment. In systems with terminals the data

---

1) Lyon, 1971, pp.11-13.
2) Parker, 1973, p.39.
3) Traver, 1973, p.8.

Figure 7

EMBEZZLERS BY POSITION OF EMBEZZLER

Position	Number of Embezzlers*
Operations Vice-President, manager, clerk	32
Loan officer, manager	29
Teller	22
President	14
Cashier	8
Director, stockholder, officer	5
Vice-President, EDP	4
EDP clerk	3
Bookkeeper	3
Trust officer	3
Programmer	3
Computer operator	3
Systems analyst	2
Chief teller	1
Auditor	1
Vice-President	1
Proof Department supervisor	1

*Includes 17 cases of collusion

administrator's office may have its own terminal with exclusive access to certain parts of the DED/DD system. This adds to the control capability of the data administrator over entering and changing data that is the data administrator's responsibility.

In security sensitive systems the data administrator's terminal lists all changes to the system that need to be monitored. Also some systems alert the data administrator on his terminal to situations such as a user trying many access codes from the same terminal in a short time period. This may be an indication of a user trying to break a code. A timely response by the data administrator's office could apprehend the possible pass-code breaker.

In addition to special hardware the data administrator's office needs a set of programs to generate reports related to the use, operation, and monitoring of the data base. Some reports related to use of the data base were mentioned in the last section. Other reports and utility type programs are listed in Figure 8.

6.4 Policy Analyst

Thus far we have discussed the personnel required to create, maintain and monitor the data base. But personnel are also required to analyse the data base and interpret it for the user. This is specially true in data bases that store both raw data and derived data such as data bases used for planning models. The data in such cases has many analytical uses which need to be identified and interpreted to the user. Such analyses have both descriptive and prescriptive value and are important by-products of data bases used for planning models. One example of such a by-product occurred in the pilot implementation of RRPM-1. One user indicated that the greatest use of RRPM was not the projected budget but an analysis of the ICLM used for projecting the budget(1).

The need for such analysis and interpretation of data is resulting in another emerging position...what Tom Mason calls a policy analyst(2). A policy analyst's role is to analyse output as well as input. For institutions using planning models the policy analyst must also analyse existing historical data for possible use as planning variables in the planning model. In such models important examples of input data include the ICLM, average class loads, average class sizes, and average room utilisation. In addition, in the case of the ICLM and student transition matrices, the matrices must be analysed for stability and integrity. In the case of faculty load-coefficients, the data must be analysed for reasonableness and consistency. Such analyses can lead to many separate projects and models such as the Faculty Activity Analysis, the Course Load Generator, and the Space Utilisation Model. All these analyses could logically be the function of the policy analyst.

Figure 8

REPORTS AND PROGRAMS TO SUPPORT THE DED/DD SYSTEM

. Listings of DED
. KWIC/KWOC Indexes
. Listings of Directories
. Testing Data Generator
. Data Division Generator
. JCL Generator
. Implementing Hardware and Software Changes
. Data Validator
. Access Controller
. Data Definition Controller
. Data Structure Converter
. Data Storage Assigner
. Statistical Reports on Utilisation
. Comparison Routine
. Utility Routines
    . Logging
    . Editing
    . Dumping
    . Garbage Collector

---

1) Hussain and Martin, 1971, p.71.

2) Mason, 1973, pp.3-4.

7. SUMMARY AND CONCLUSIONS

In very few institutions of higher education has the top administrator devoted an adequate amount of attention to the organising of information systems. While it should not be expected that administrators become qualified in the technical aspects of information systems, it is important that they be concerned about the control and management aspects. The costs of a successful information systems operation are considerable. The costs of unsuccessful operations are even higher.

It has been the purpose of this chapter to acquaint the institutional administrator with some of the factors involved in organising information systems. The organisational roles in both system design and use were discussed. The interrelated systems concept and a model for a user's group were presented. The internal organisation of the data processing organisation was covered; the major functional responsibilities of that organisation discussed; the attitudes of both the user organisation and the data processing organisation were identified; and finally, the role of the data administrator and policy analyst were analysed.

The many users and creators of data (manual and computerised data) involved in the development of the DED were identified. They must be controlled so that necessary standards of reliability and integrity of data are maintained. This control is usually performed by a data administrator (a position which only recently has evolved) under the supervision of a Users' Committee. This chapter outlined the qualifications needed in a data administrator, his role and responsibilities. Also discussed in this chapter was another new position, that of a policy analyst. He interprets data and output information for its policy implications.

The DED/DD system can be used by institutions at all levels of sophistication in the development of their data system. The most basic use would be to use the DED as a repository of definitions of data elements. As the data base gets larger and more complex, directories and DED/DD software could be added, evolving the system into a Data Base Management System.

An important part of the organisation of an information system is the funding of the information system, and the related topics of what is the cost and benefit of information. This raises many issues that can be categorised as economic aspects of an information. These are the subjects of the next chapter.

REFERENCES

Albrecht, L.K., Organization and Management of Information Processing Systems. New York: The Macmillan Company, 1973, p.183.

Caffrey J. and C. Mosmann, Computers on Campus. Washington D.C.: American Council on Education, 1967, p.183.

Cahill, J.J.,(1970), " A Dictionary/Directory Method for Building a Common MIS Data Base", Journal of Systems Management (November), pp.22-29.

Hussain, K.M. and J.S. Martin,(1971), A Resource Requirements Prediction Model RRPM-1: Report on the Pilot Studies (Boulder, Colorado: NCHEMS at WICHE).

Lyon, J.D.,(1971), "The Role of the Data Base Administrator", Data Base.

Mason, R.R.,(1973), "The Purpose of Analytical Models: The Perspective of a Model User", Paper presented at the CSCUP Conference.

Mosmann, C., Academic Computers in Service. San Francisco: Jossey-Bass, Inc., 1973, pp.66-71.

Parker, D.B. et al., (1973), Computer Abuse (Menlo Park, California: Stanford Research Institute).

Taylor, R.E., "Developing Computer-Based Management Information Systems". Organizing for Innovation: Towards the Computer Utility. Conference Proceedings, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, January 17-19, 1973, pp.39-70.

Thomas, C.R., "Data System Design". Prepared for Program on Institutional Management in Higher Education, sponsored by Organisation for Economic Co-operation and Development, Paris, France, October 1-5, 1973.

Traver, C.M., (1973), "Data Base Administrator - The Emerging Position", Proceedings of the College and University Machines Records Conference.

Willoughby, T.C., (1972), "Staffing the MIS Function", ACM Computing Surveys, Vol.4, No.4, pp.241-259.

Chapter 10

THE ECONOMICS OF AN INFORMATION SYSTEM

by

James Farmer
Systems Research Inc.,
Oklahoma, U.S.A.

CONTENTS

1. Introduction 194
2. Utilisation 194
3. Differential Pricing 196
4. Economy of Scale 197
5. Risk 201
6. Investment Mix 202
7. Management Action 204
8. Programmer Productivity 204

References 210

1. INTRODUCTION

The economics of data processing differs from some of the more conventional manufacturing processes, and these differences may lead to non-rational economic behaviour. This chapter explores some of these concepts and relates them to cost and benefit data from applications in colleges and universities.

The concepts include the familiar utilisation, risk, investment, and operational costs and cost-benefits.

2. UTILISATION

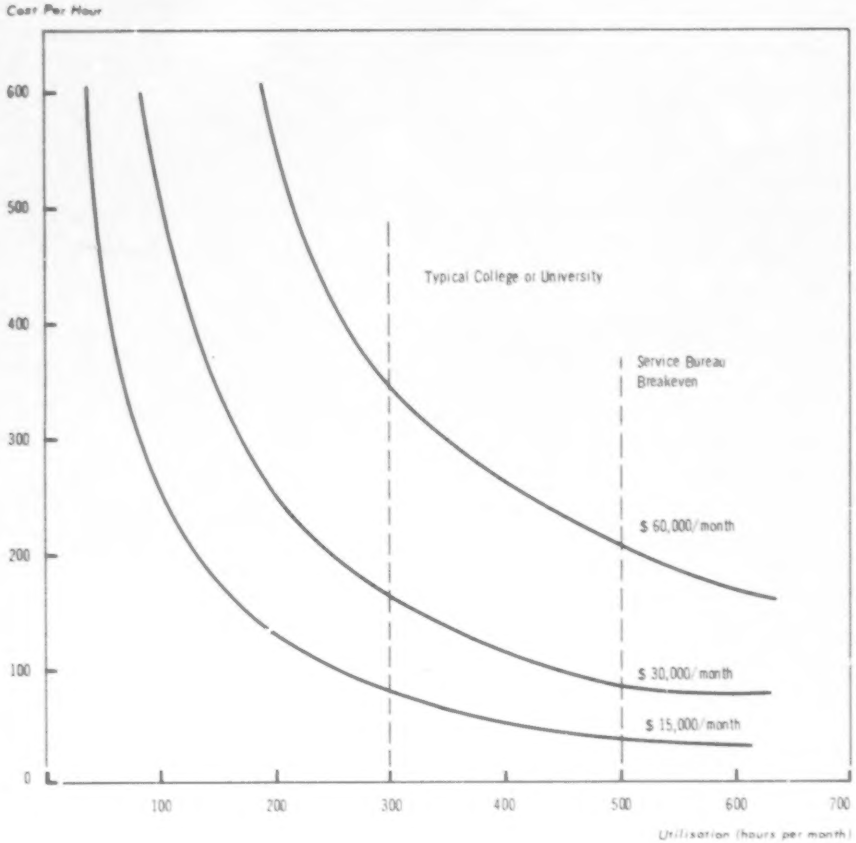
The ultimate user of a data processing service is not purchasing computer hardware or software, but rather is purchasing some processing capability. The price is often not known; it is buried in some complex algorithm. Benchmarks to determine true costs and prices are rarely used in colleges and universities; they tend to be difficult to construct and even more difficult to interpret.

For purposes of investigating the effect of utilisation of a computer configuration on prices to the user, consider the situation where all users combined must bear the cost of the computer installation - the most frequent case is those colleges and universities with chargeback systems. Assuming that the costs of a computer centre are approximately twice the rental cost of the equipment, then these hourly costs are shown in Figure 1. To aid in interpretation, the breakeven point of most commercial service bureaus and a typical utilisation of a college and university computer are also shown. Three sizes of computers are shown, those renting for $30,000, $60,000 and $120.000 per month.

From these rates it is possible to observe what happens when a computer centre begins to lose applications. For example, if the average utilisation fell from 300 hours per month to 200 hours per month, the cost per unit of computation would increase by 50 per cent. Although some colleges and universities do not exhibit rational economic behaviour, many college professors will attempt to use alternative computing resources if prices exceed the market by more than, say, 100 per cent. Now, as the utilisation decreases, prices increase; utilisation further decreases, and prices further increase. Thus, a computer centre which freely competes with alternate sources of computing service is meta-unstable. As utilisation begins to

Figure 1

COST PER HOUR BY UTILISATION



decrease, the costs increase rapidly and soon there are no users. Similarly, if there is an increase in utilisation, prices go down. Thus, more use is made of the computer and prices further decline.

There is, however, one factor which tends to stabilise computer prices. As the utilisation decreases and prices increase, service, measured in turnaround time or job queue length, will improve; and, similarly, as utilisation increases, queue length becomes longer and approaches infinity as the utilisation moves close to 720 hours - the maximum available time.

From experience, the user community tends to be stable from 300 to 500 hours per month and then service considerations dominate costs beyond these limits.

The conclusions are clear. The computer selected for an institution should be small enough to maintain the utilisation between 300 and 600 hours.

3. DIFFERENTIAL PRICING

There are, unfortunately, only 76 prime hours per month, and it is difficult to achieve a high utilisation if all users attempt to process only during the prime hours. For a university, these tend to be from 10.00 to 18.00 each day with sharp peaks at 11.00 and 14.00. Apparently, these peaks are the result of class scheduling as well as the noon hour and habits of faculty and students.

Because of an interpretation of a rule for federal government reimbursement, most U.S. colleges and universities tend to charge their users the same rate regardless of the time of use. There is, of course, no price incentive for a user to change his demand for computer services to non-prime hours. With the advent of heavy keyboard terminal use, the load on the computer centre became very heavy during the prime shift, and very light during the non-prime shift. Again, the only reason for demand at other than prime times came from the improved service when there were fewer users on the system. Some systems have keyboard response time (from time of the carriage return to the first character returned) during prime hours from 10 to 20 times that of non-prime hours.

There are two ways of improving computing service and cost. If the computer centre has any direct control over the workload, then portions of that workload will be shifted to the non-prime hours. Thus, typical administrative applications and lengthy research problems can be shifted to late evening and early morning hours. If the computer centre does not have direct control, then it can use pricing to encourage non-prime use. The American Council on Education in the United States offers a 40 per cent discount to users after 5.00 p.m.

and before 8.00 a.m. Prime shift on-line users have a higher priority and a 100 per cent surcharge for computer central processor time - approximately a 40 per cent increase. Thus, the difference between prime and non-prime use is about 50 per cent. For this amount of differential, the American Council on Education has been able to shift about one-third of its keyboard terminal workload to non-prime hours and about one-half of its batch work. This shift improved service to prime shift users considerably. The University of Illinois recognised that its administrative computer centre, heavily on-line keyboard terminal user-oriented, would give a large unused capacity at night. By special rate reductions for research problems which are heavily computer-bound, and which could be done in any of several days, the university has one of the lowest rates in the United States with a utilisation rate of about 90 per cent. At the same time, the computer-bound research jobs are entered at the lowest priority which does not adversely affect service to the on-line terminal users. At the same time, it provides extremely low research computing costs. Because of the low prices, the university has been able to provide large amounts of computing for those research areas, such as crystal rotation, X-ray defractions, and genetic simulations, which otherwise might be too expensive to adequately support. Benchmarks show that the University of Illinois is about 50 per cent less than the most competitive on-line terminal service in the United States, and up to 75 per cent less for low priority research computing. The key, of course, is the high utilisation resulting from a bold policy of price differentials and imaginative computer workload scheduling.

4. ECONOMY OF SCALE

Significant economies of scale are supposed to accrue from large computer installation. The so-called "Grosch's law" claims that doubling the monthly rental should increase the capability by a factor of eight. Some typical benchmarks of student jobs did not support this assertion. Table 1 shows the costs of a typical student Fortran benchmark on different classes of machines. These benchmark results include both Control Data Corporation and IBM equipment(1). Because most users are familiar with IBM models, the computers are identified by class. The typical student benchmark varied from $3.20 on a 30 class machine to $1.09 on a 50 class machine. For purposes of analysis, the compilation and linkage-edit were listed separately from the computation.

---

1) Farmer, Harrison and Smith, 1972.

Table 1

STUDENT JOB BENCHMARK COSTS

Class of Machine(1)	Compilation, Linkage	Computation
30	\$1.60	\$1.60
40	\$0.90	\$0.60
50	\$0.61	\$0.48
65	\$0.60	\$0.32
90	\$1.93	\$0.04

1) Includes both CDC and IBM machines

These results are shown graphically in Figure 2. The minimum price was \$.92 on a 65 class machine with both the 90 class and 30 class machine costing more. Since there has been an understanding that larger computers are more efficient than smaller computers - definite economies of scale - it appears unreasonable that student jobs would cost more on a larger machine. There are three reasons for this. First, compilation and linkage-edit, as contrasted to computation, requires a significant amount of input-output operations and hence uses a large computer inefficiently (compared to lengthy computation) and yields higher prices. Similarly, at the low end of machine size, the computation becomes much more expensive. Second, it is much more difficult to maintain high utilisation on a large machine than on a smaller machine. The speed of the equipment and size of the installation makes the physical job handling much more difficult and user demand much less amenable to change. Third, there appears to be more operating system overhead for a larger system than for a small or medium scale machine.

There is some evidence that even better cost-effectiveness can be obtained from smaller 40 class machines which are specifically configured for student jobs - high speed card reader and printer, and minimal core and storage. The shape of the cost curve is significant: it suggests that distributed networks, like the California State University, would be more cost-effective than large central installations.

In order to have better insight into costs, the cost of a large scale computation - three dimensional matrix manipulation - was determined by benchmark. The results are given in Figure 3. The cost decreases markedly with scale. Since the saving is substantial - the 90 class saves some 75 per cent over the 30 class machine - this suggests large central computational facilities even if there is a lower overall utilisation. Usually the economic argument for centralisation is increased utilisation by combining workload rather than the lower per unit computation resulting from a larger scale machine.

Figure 2

CLASS OF MACHINE

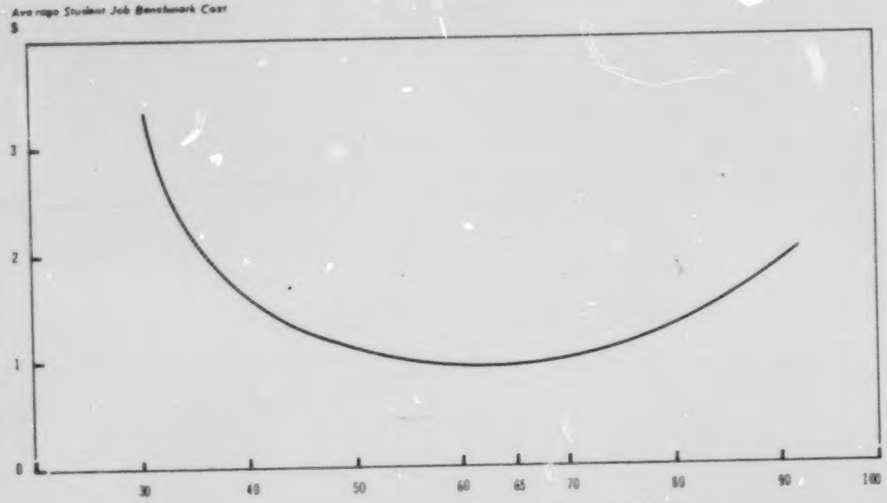
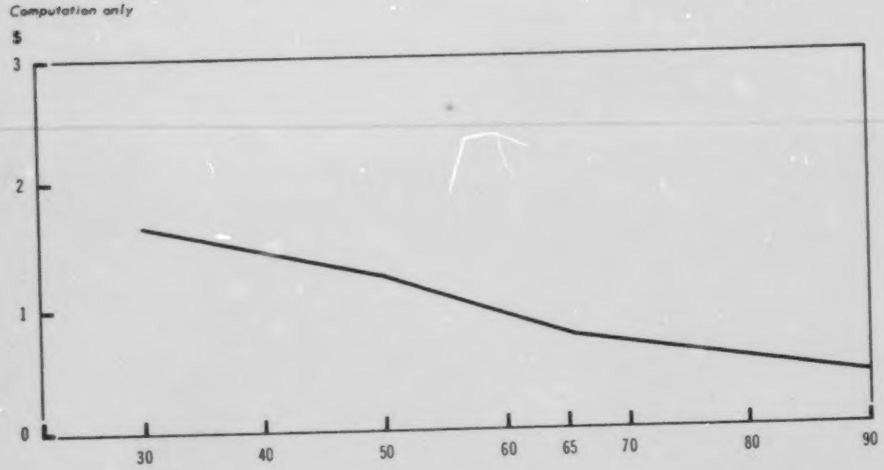


Figure 3

CLASS OF MACHINE



In order to see a comparison between a large academic installation and a commercial installation, the benchmarks were run on a large, heavily subsidised computer centre in a major metropolitan area, and a commercial service bureau located only a few blocks from campus. The results are shown in Table 2. The commercial installation was competitive, and for the short benchmark problems, was less. Since both installations were using IBM's OS release 20, the overhead should be grossly comparable. The primary difference is in utilisation. Based on use (i.e., time when there are jobs in the queue), the commercial installation is used approximately 450 hours per month compared to some 250 for the academic computer centre. Thus, the profit-making commercial centre offers reduced prices even with the large subsidy a major academic computer centre has. These benchmark comparisons used IBM's Fortran G.

In order to see the effects of software, Table 3 shows the differences between Fortran G, Fortran H with level 02 optimisation, and WATFIV. Three benchmark options were used: Input/Output - reading 200 cards and formating a 1,000 line output; arithmetic - performing 100,000 arithmetic operations on a three-dimensional matrix; and, in order to determine the total overhead of compilation, linkage-edit and initiation and termination, a benchmark which merely called for programme exit. Table 4 shows the results of the input/output or arithmetic processing itself. While these results are not conclusive because of the restricted size of the benchmark, there are some obvious trade-offs which must be considered. Software offers both different quality - error messages for students, or optimisation for large research tasks - and different costs for identical executions. Subsequent benchmarks tend to support these conclusions. It appears that both management policies and technical management decisions - selection of software, priorities, configuration, and work flow - significantly affect the unit cost.

Table 2

COMPARATIVE PRICES

	Large Academic(1)	Medium Commercial
Fortran Compilation	$0.68	$0.79
Fortran Arithmetic		
Compile, Link Edit	$1.93	$1.51
Execution Only	$0.04	$0.32
Fortran Input/Output		
Compile, Link Edit	$1.93	$1.51
Execution Only	$0.17	$0.95

1) The large academic centre uses 4 megabyte IBM 360 Model 91 with at least 50 per cent subsidy. The medium commercial service bureau has a 1 megabyte IBM 360 model 50. Both centres offer approximately the same consulting systems engineering support.

Table 3

STUDENT JOB BENCHMARK COST-SINGLE INSTALLATION

	Input/Output(1)	Arithmetic	Enter/Exit
Fortran G	$2.10	$1.97	$1.93
Fortran H (02)	$3.02	$2.85	$2.74
WATFIV	$0.87	$2.31	$0.54

1) Compilation, linkage-edit, and execution.

Table 4

STUDENT JOB BENCHMARK COST

	Input/Output(1)	Arithmetic
Fortran G	$0.17	$0.04
Fortran H (02)	$0.28	$0.11
WATFIV	$0.33	$1.77

1) Execution only. For Input/Output, 200 cards read, 1,000 lines of formated output. For Arithmetic, an average of 100,000 fixed or floating, additions and subtractions or multiplications and divisions using a three-dimension matrix stored as block data.

5. RISK

A user would usually like to pay for a service as it is consumed, and often would prefer to lease equipment rather than purchase the equipment. Several years ago, manufacturers offered lease arrangements which were similar to amortising the purchase over 48 to 60 months. Essentially, the manufacturer was absorbing the risk of obsolescence. Since major manufacturers could accurately predict the rate of introduction of their new products, it was appropriate for the manufacturer to bear this risk. Leasing companies recognised, however, that it was possible to extend the life of equipment beyond the typical 48 to 60 months, and purchased the equipment and offered it to users at less than the current manufacturer's lease rates. This was viewed as being a bold move in the late 1960s when most users were purchasing each new series as it was introduced, and frequently changed equipment for larger sizes every two or three years. When IBM began deliveries on the System 370, it looked like the leasing companies would have obsolete equipment. As it turned out, both IBM's System 360 and machines of similar technology by other manufacturers have been productive for the past decade as new peripherals and software modifications have increased performance.

The decision to lease versus purchase depends upon two factors: the cost of funds, and the risk of obsolescence. If the university, the manufacturer, and leasing companies are assuming similar useful life for equipment, then the issue is one of simple investment analysis - is it preferable to lease or purchase? If, however, the useful life to the university is less than assigned by a leasing company or manufacturer, then the university should lease the equipment if it can transfer the risk of obsolescence - that is, obtain a short-term lease - to a third party. Since a market has developed for used computers, and there is a market for the second generation of lessor on some equipment, it may well be possible to effect a lease.

6. INVESTMENT MIX

Most administrators or managers find themselves giving careful attention to the cost of the computer hardware and relatively less regard to administrative application software. Yet a study of computer installations of the IBM Model 30 class demonstrated that the investment for software to achieve a 500 hour utilisation was approximately that of the hardware itself. If a 50 class machine cost about twice as much, and the economies of scale were about one-half those predicted by Grosch, then the software investment would be twice the cost of the machine. By a class 75 machine, the cost of the software for administrative applications would be only 11 per cent of the total hardware and software investment, as shown in Figure 4.

Because many college and university computer centres service faculty and students who either write their own programs or use prepared programs, these observations would not hold for combined administrative and academic computer centres. However, for the administrative computing centre these data appear to hold. Thus, two phenomena occur - the utilisation is low until a major investment has been made in administrative software, and the software investment far exceeds the hardware investment. This may be particularly difficult for a college or university to bear since the hardware costs are usually carefully budgeted, but the cost for systems development is rarely estimated accurately.

There are, however, strategies for extending these costs across several users. Development of administrative applications by consortia would tend to reduce the costs. Assuming three users participated equally in a consortium and the final user spent as much for the modification as was spent for the development, the results would be a 33 per cent decrease in the combined hardware and software development(1). If a packaged system is used, assuming the investment is

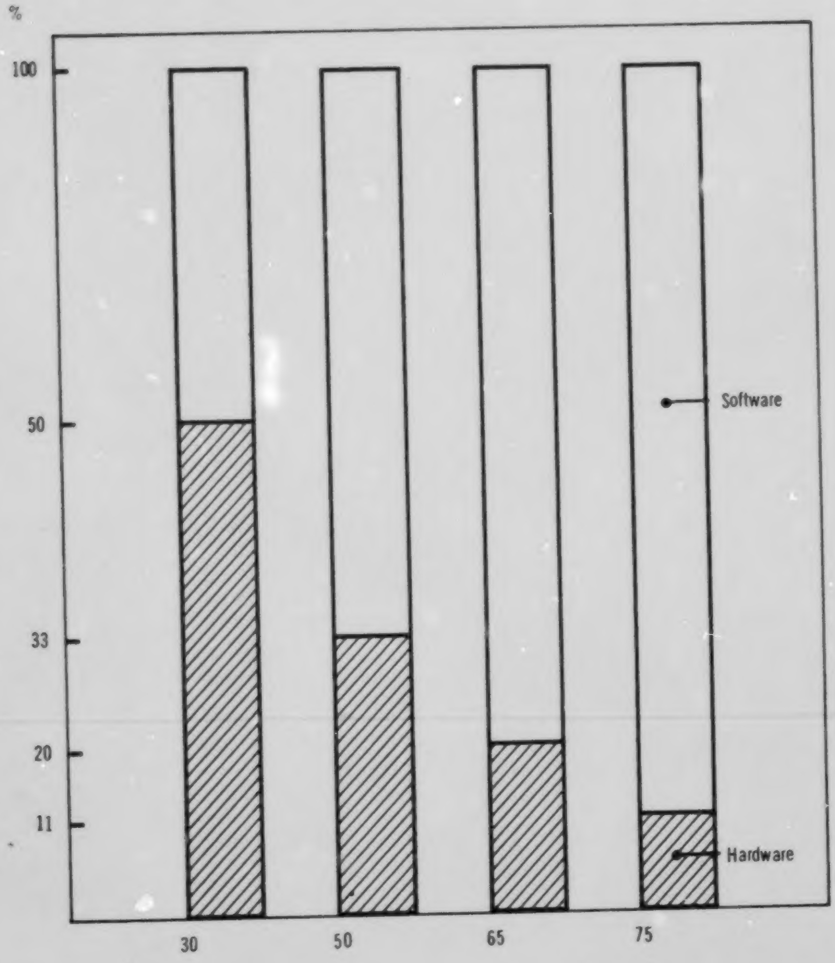
---

1) Chauncey and Greenberger, 1971.

Figure 4

CLASS OF MACHINE

Relative Investment in Hardware and Software for Hour Utilisation



spread among ten users and that each user spends as much modifying the package as it costs, the result is a software and hardware investment only 45 per cent of the cost of developing all applications by the organisation. These relative costs are shown in Figure 5.

7. MANAGEMENT ACTION

Computing was a product of college and university research. For a period of time computing was a symbol of status among institutions. Now computers are merely professional tools needed by most researchers, an object of instruction, a learning aid for students, and an alternative method of administrative data processing. Because of the origin of computing, many college and university administrators view computing in its historical rather than current role. However, administrative data processing can be viewed as any other college investment, and a mixture of hardware and software can be obtained to satisfy the administrative requirements at minimum cost. Administrators must recognise the effects of economies of scale, risk, investment mix and pricing in order to achieve a viable and stable computing service.

8. PROGRAMMER PRODUCTIVITY

For a long period of time there has been an assumption that programmer productivity was linear. That is, if a programmer could produce 3,000 lines of COBOL program in one month, then a program of 6,000 lines could be produced in two months. Our experience now shows that longer programs are usually more complex or, at least, require the programmer to be more attentive to the decreased productivity which results.

Similarly experience has demonstrated that the environment in which a programmer works significantly offsets his productivity as measured by the number of instructions or lines of coding. As the number of instructions increase, productivity declines.

It may be useful to review the data before drawing some general conclusions. Figure 6 shows the number of COBOL statements, including the data division, which is produced. A typical 2,000 line program can be produced in 2 weeks or similarly two programs of 1,000 lines in the same system may be produced. As the size increases, say to 4,000 lines, the time required does not double, but increases to approximately six weeks - some three times the amount of time. The time required to complete the programming increases rapidly as size increases. Apparently, a programmer's productivity is closely related to the information he must remember in the process of developing and testing the remitting computer program.

Figure 5

CLASS 50 MACHINE

Relative Investment in Hardware and Software for Software Development Alternatives

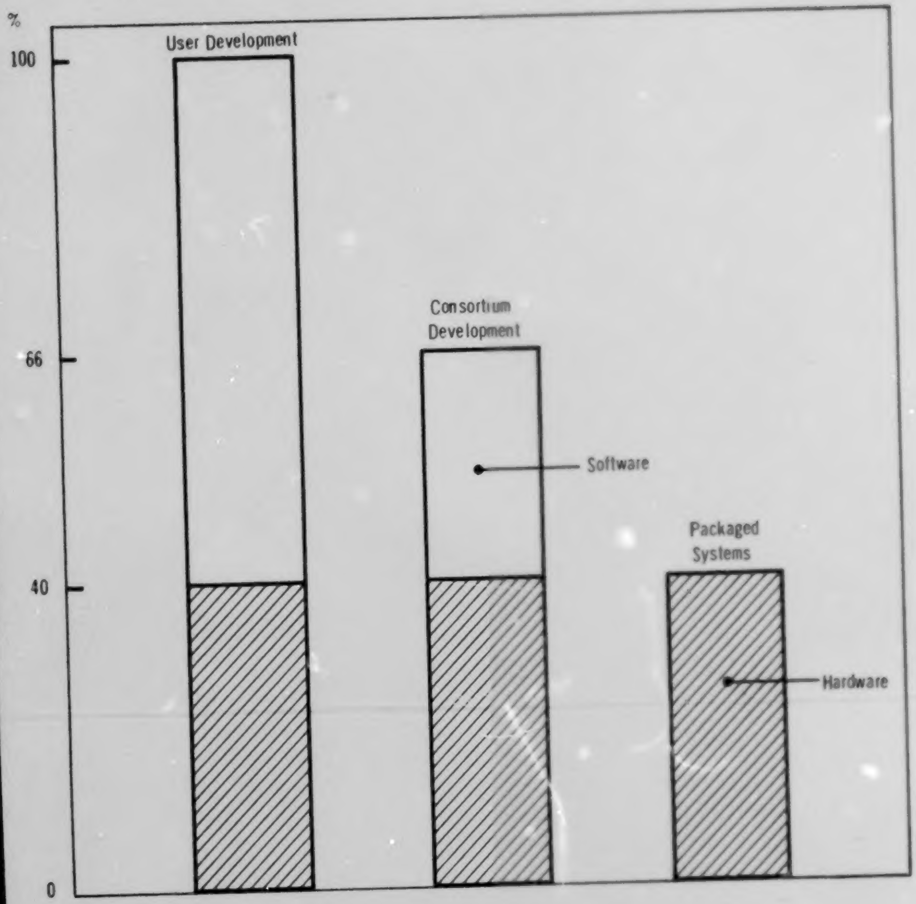
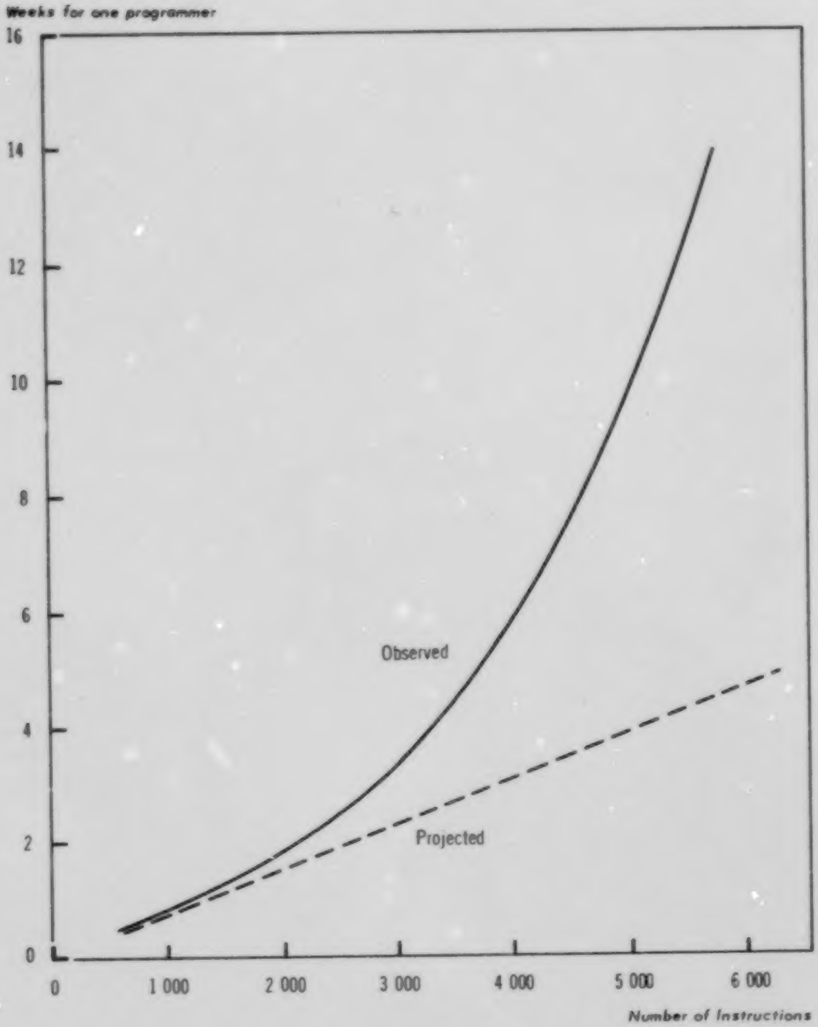


Figure 6

PROGRAMMER PRODUCTIVITY AS A FUNCTION OF SIZE

(Small System)



Similar results have been demonstrated for large systems. The results are shown in Figure 7 where 250 thousand lines of code require 1,500 man months while 500 thousand lines require 4,200 man months - about three times as much. While few colleges and universities will embark on projects of this size, the principle is clear - large projects require proportionately more programming effort and there may be a serious underestimate of programming effort by those who expect linear relationships.

Figure 8 shows a different relationship as the number of lines produced by a programmer is related to the interaction between a programmer and the environment. If a programmer has a number of interrelationships - direct contacts with users, liaison with many other programmers - his productivity is low - here approximately 1,070 lines of program code or output. If the programmer has complete specification with no interaction, he produces 15,000 lines. While these results apply to large systems, it suggests that design uncertainty, or a design which requires close co-ordination between programmers, will significantly reduce potential programmer productivity.

If these observations are generally applicable, then modular design, even those which require operating inefficiencies may significantly improve system designs. Since a programming productivity increase of 50 per cent to 100 per cent seems possible, the operational cost increase of 5 to 15 per cent seems reasonable. The exact trade-off between development investment and operational costs depends upon the specific application.

Also, it seems reasonable that increased design effort may facilitate programming. Then it may be a wise strategy to significantly expand the design effort to reduce uncertainty in the design or any potential area of co-ordination which may traditionally have been left to the programmer.

Figure 7

PROGRAMMING PRODUCTIVITY AS A FUNCTION OF SIZE
(Large Systems)

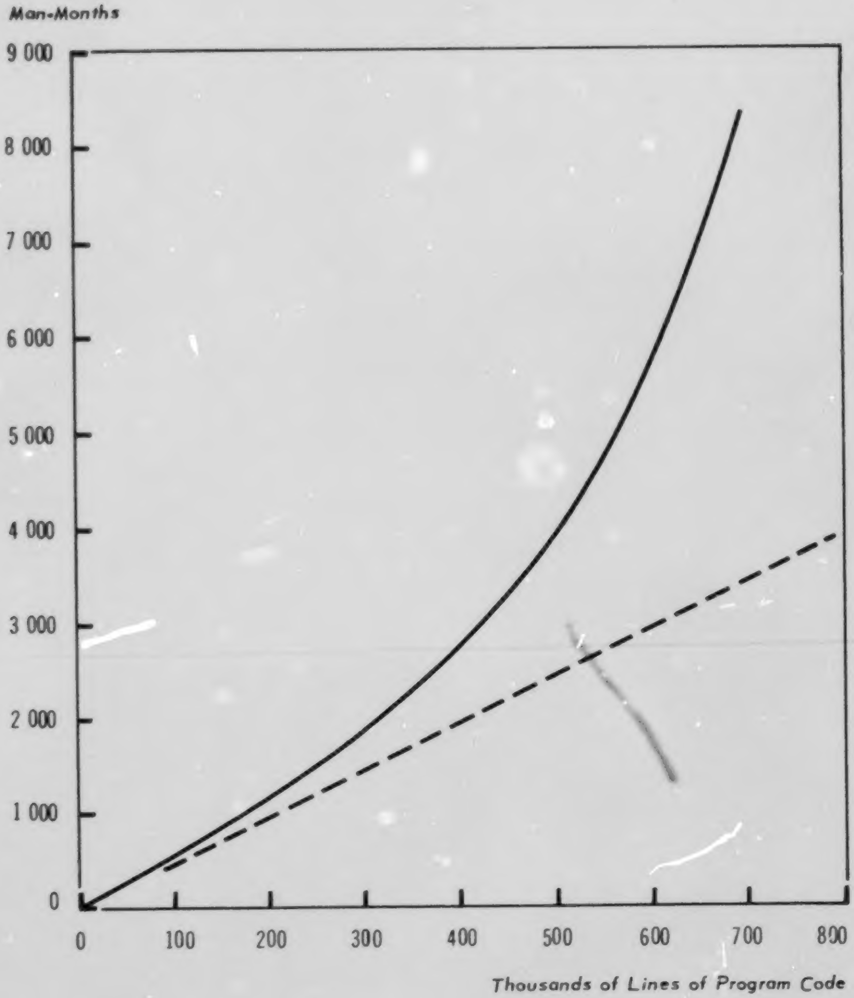
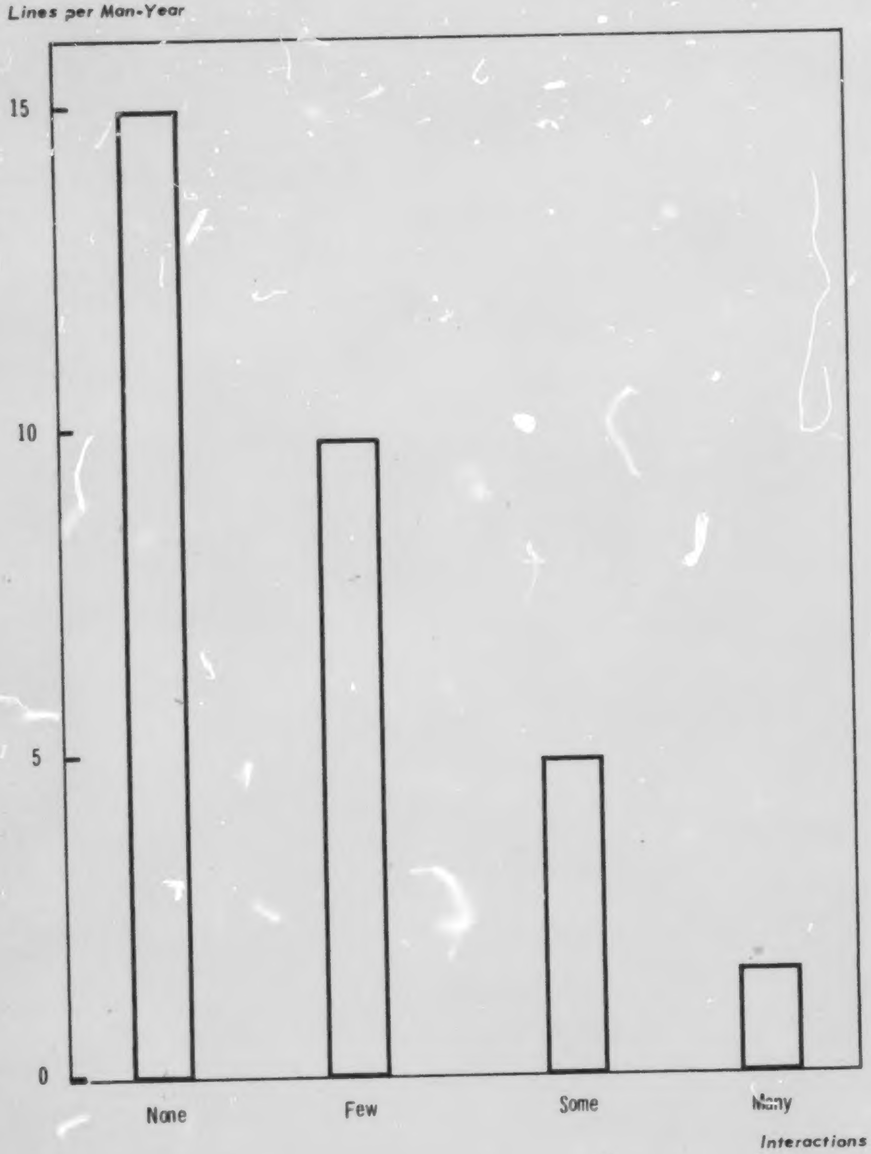


Figure 8

PROGRAMMER PRODUCTIVITY
(Large Systems)



REFERENCES

Farmer, James, John G. Harrison and Mark C. Smitt, "The Economics of Computing in Higher Education: Hardware/Software Performance". Proceedings, College and University Machine Records Conference, May 1972, The Ohio State University, Colombus, Ohio, May 1972.

Chauncey, Henry and Martin Greenberger (ed.), The Financing and Organisation of Computing in Higher Education: 1971, EDUCOM Interuniversity Communications Council, Inc., Princeton, New Jersey, 29th April, 1971.

Part Four

CASE STUDIES ON INFORMATION SYSTEMS IMPLEMENTED IN IMHE INSTITUTIONS OUTSIDE THE U.S.

This Part consists of four case studies on information systems implemented in IMHE institutions outside the U.S. They are presented here partly because such material is not easily available in the literature and partly because they complement with application examples the conceptual material on information systems presented in the preceding pages.

The cases were selected largely on the basis of the ratings of a judgemental sample of participants of the IMHE/OECD seminar where these cases were presented. Also, the cases were selected because they offer a variety of applications. They include both an application in a new institution as well as an application for a "mature" institution; application of a single operational function as well as that of an integrated comprehensive system; and finally, the applications in both small as well as large complex institutions.

Specifically, the four cases selected for the following pages are on: designing and implementing a data base; an automated library check-out system; a course management system; and an integrated system for university management. These cases will now be introduced briefly so as to help the reader in selecting the case(s) of interest.

The first case is on the development of a data base at the Universidade Nova in Lisbon, Portugal. At the time of the design and implementation of this project, the university was expecting its first large enrolment three years hence. Therefore, developers of the system were able to start from scratch without the usual problems of restructuring in an ongoing institution. There was, however, one main disadvantage: the information needs of the university did not exist. This and other unique problems of new institutions are examined. Problems common to all institutions, such as those inherent in any development cycle as discussed in Chapter 3, are also examined along with the basic concepts of data base development discussed in Chapter 4.

The second case is on an automated book issue system at the University of Bath in England. This is a functional application and is a subsystem of the information system for the library. It is conceptually a simple system because it has none of the problems of the integration since the library system is largely isolated and not greatly integrated with the other systems in the university. But it has a high benefit-cost ratio in its operation even without including the benefits derived by the student and faculty user. However, in spite of the conceptual simplicity and economic effectiveness of this subsystem (as well as some other library subsystem), it is not a very common or popular application in universities. This is partly because it does not have the conspicuous benefits of a student information system; nor does it affect every university employee as does the payroll system Nonetheless, it should be examined by every university manager for feasibility and cost-effectiveness. It is especially cost-effective in countries where most published books are assigned an identification code appropriate for computer processing and in countries where many professional journals require that the author provide an abstract and a set of keywords for his article. These can be used as direct input to an SDI (Selective Dissemination of Information) System which matches the keywords for each article with the user's profile of keywords. It then identifies articles of likely interest to each user. This information can be valuable not only to faculty and researchers but also to university managers.

The third case is on Course Management at the Laval University of Canada. This case is an example of integrating three subsystems in the system of course management. The three subsystems are: one, course information required in announcements of university offerings to new students considering admission; two, course information needed at the time of student registration; and three, course information related to the students' academic record at the university. All these three types of needs of course information are integrated in one course management system at Laval. It is not a trivial system, since the system is large (about 10,000 new students seeking admission and 20,000 students seeking registration every semester) and very complex (100,000 courses offered in 350 different curricula by 60 departments).

The fourth and final case is on an integrated management system at the Katholieke Universiteit te Leuven in Belgium. This case relates to the integration of all systems in the university. It uses the common data base approach and the advanced techniques of data base management discussed in the latter part of Chapter 4. Problems of applying these advanced concepts in the real world are discussed.

It is interesting to note that the advanced and sophisticated techniques of data processing used in this case were successfully implemented in a relatively small university (10,000 full-time equivalent students) and in a small country (Belgium).

The technical details of this case (as well as with the other cases reported earlier) have been deleted because they would be of little interest to the manager. Such details are, however, available in the documents published for the seminar or can be obtained directly from the authors.

Chapter 11

DEVELOPING A DATA BASE FOR A NEW EDUCATIONAL INSTITUTION: A CASE STUDY OF UNIVERSIDADE NOVA DE LISBOA

by

K.M. Hussain(1)

CONTENTS

1. Introduction 216
2. Overview of Project 216
3. Work of the Data Team 222
4. Work of the ADP Group 228
    4.1 Manpower Needs 230
    4.2 Equipment Needs 230
    4.3 Scheduling Computer Output 232
    4.4 Designing System 232
    4.5 MIS Plan 232
5. Resources for the project 232
    5.1 Time Required 234
    5.2 Manpower Resources 234
6. Summary and Conclusions 234

---

1) The author was a Fullbright Professor at the Universidade Nova de Lisboa, Lisbon, Portugal in summer 1974.

1. INTRODUCTION

In 1973, the Government of Portugal authorised the founding of the Universidade Nova de Lisboa with its first set of regular students in 1976. A Rector was appointed with staff for planning and organising the new university. In summer 1974, the work on budgeting and long-range planning was reviewed. It was determined that the university was well advanced in its conceptual development of planning and budgeting models but there was no work done on the data base (set of data required) or the planning of a management information system (MIS). The data base and MIS was required not only for budgeting and long-range planning but also for daily operations and management.

Since some expertise in this area was available for Summer 1974, it was decided to institute a crash project to develop an institutional data base and a plan for an MIS. The work of this project is the subject of this chapter. It will describe the different activities (tasks or jobs) and their interrelationships in Sections 2, 3 and 4; will identify the resources required in Section 5; and finally discuss some lessons learned and conclusions reached in Section 6.

This chapter is addressed primarily to the administrator in higher education. It will answer questions like: what type and amount of personnel will be required? What other resources will be needed for the project? What types of decisions have to be made? How long will it take? The answers to these questions should be of interest to administrators of all types of institutions, but especially to the new institution. Their unique problems and advantages in building a data base will be identified and analysed; what will not be done, however, is to develop basic concepts of data or data processing or project management. This has been done elsewhere(1) and will not be repeated here.

2. OVERVIEW OF PROJECT

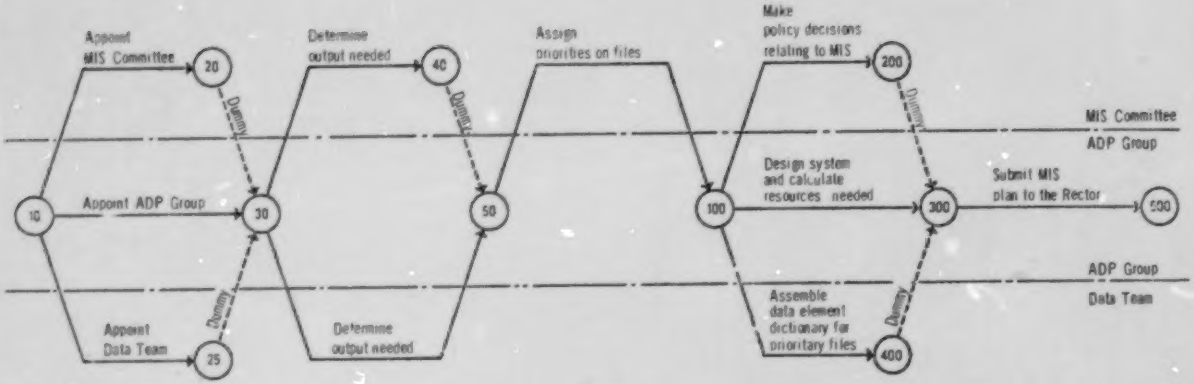
The overview of the activities of the project is shown in Figure 1. It starts with the appointment of three ad hoc working

---

1) For a reference of project management including CPM and PERT (a technique used in this paper) see Handy W. and K.M. Hussain (1969) Network Analysis for Project Management, Englewood Cliffs, U.S.A., Prentice-Hall Inc., 186 pages. For data processing specifically addressed to the educational manager, see Hussain K.M. (1973) Development of Information Systems for Education, Englewood Cliffs, U.S.A., Prentice-Hall Inc., 419 pages.

Figure 1

Overview of Project


Appoint MIS Committee
20
Dummy
Determine output needed
40
Dummy
Assign priorities on files
Make policy decisions relating to MIS
200
Dummy
MIS Committee
ADP Group
10
Appoint ADP Group
30
50
100
Design system and calculate resources needed
300
Submit MIS plan to the Rector
500
ADP Group
Data Team
Dummy
Appoint Data Team
25
Determine output needed
Assemble data element dictionary for prioritary files
400
Dummy

Note : Dashed arrows are «Dummy» arrows to facilitate unique identification of all activities.

groups: the MIS Committee (activity 10-20); the ADP (Administrative Data Processing) Group (activity 10-30); and the Data Team (activity 10-25).

The MIS Committee was composed of middle management and represented with users. They were responsible for all policy-type decisions and would consult with the Rector whenever necessary. The ADP Group consisted of technical personnel knowledgeable in data processing. It was responsible for technical problems of design and of estimating costs of the future system. It was assisted by the Data Team, consisting of analysts who were to help define the output and the data elements. All personnel were appointed by the Rector from his existing staff, except for a consultant from the United States, who was a Fullbright scholar assigned to the Universidade Nova de Lisboa for the Summer.

The work of the ad hoc groups is shown in Figures 1, 4 and 8. In all cases, the activities are represented in one of three levels horizontally, a level corresponding to each of the three ad hoc groups.

The first task of the working groups was the determination of the needed output by the MIS Committee (activity 30-40) and the Data Team (activity 30-50). The latter activity was partly undisputed as was the case of output required by the government of all educational institutions. Other output was anticipated by management including the MIS Group.

Finally, output was recommended by the ADP Group which continually advised management on what was technologically possible and was involved in generating different outputs.

Once the output was determined, priorities were assigned by the MIS Group (activity 50-100) to the different data files (sets of data). This was done on a special form and then attached to each output for the record. A sample is shown in Figure 2. Based on the priority assignments, a schedule of files was prepared and is shown in Figure 3.

The files shown are physical files. They could be combined. For example the Student and Admission file could well be combined. Its desirability would depend on the data processing equipment available and the design of the System. But the priorities assigned did "distribute" the development work evenly over the five year development period. This was quite unexpected.

Typically, there is a strong demand for developing all files, either because of a genuine or perceived demand; or because of the need of the prestige and symbol attached to automation. In the case of the Universidade Nova (and most new institutions) there were two reasons that such a simultaneous demand did not develop. One was that all the administrative departments had not been started. For example there was by 1974 no department of Admissions nor a Business Office. A Business Officer was to be appointed soon and yet his Finance and Accounting File was postponed till four years later in order to give the system time to stabilise before it is automated.

Figure 2

SAMPLE OF DECISION FORM ON OUTPUT

UNIVERSIDADE NOVA DE LISBOA

PROCESSAMENTO DA DADOS DA GESTAO - OUTPUT

NOME DO OUTPUT: Alunos requereutos do admissao

DATA DA DECISÃO: 23/7/74

	Afonso	☒	Dra. M. Emilia	☒
PESSCAS QUE TOMAM A DECISÃO	Arg. P. Parata	☒	Marie Luis	☒
	Dra. M. Teresa	☒	Luis	☒

PRIORIDADE	ANO	78/9	MUITO IMPORT	☒
	ANO	77/8	NECESSÁRIO	☒
	ANO	☐	DESECÁVEL	☐

RAZOES: For reporting to government

OBSERVACÕES: Needs to be reviewed by Admissions

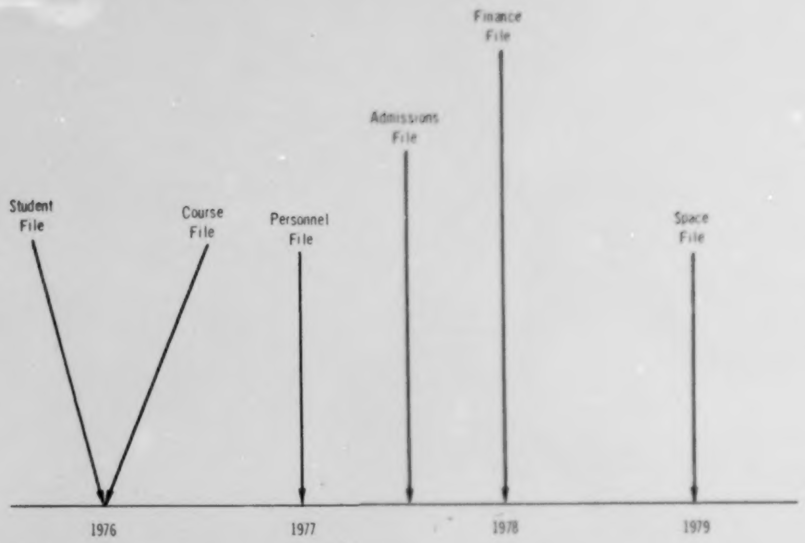
Officer on his appointment

and integrated in his file

The second reason was that one department though in existence had no need of a file till five years later. This was the department of Buildings and Space Planning. They did not expect their permanent buildings till 1979 and hence were willing to postpone their file till then.

Figure 3

Schedule of file production

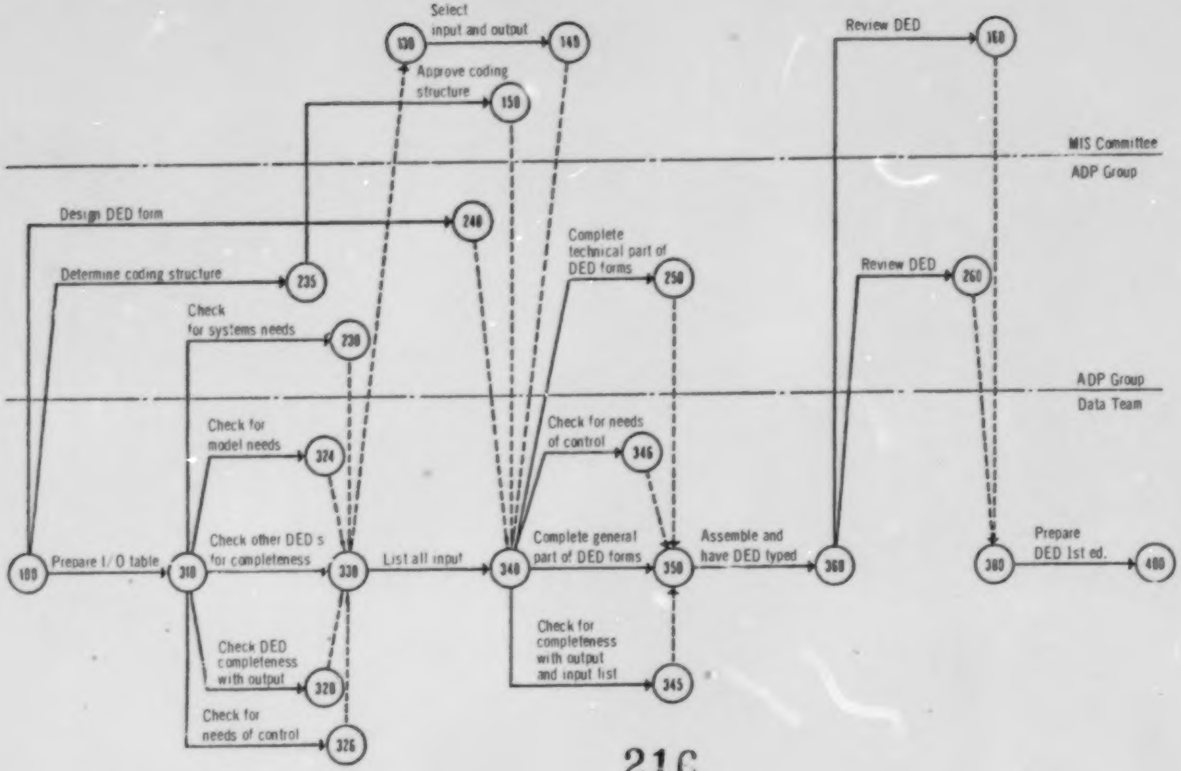


There were still some demands for a simultaneous development of all files but this was resisted partly on the grounds of not automating a system not yet conceived let alone stabilised; and partly on the grounds that a gradual development of the system was necessary to allow for "learning" about an MIS, both among the analysts and among the users.

Once the priority of files are determined (activity 50-100), then all the three working groups must work in real earnest. The Data Team must assemble the Data Element Dictionaries (to be explained later) for the priority files (activity 100-400); the ADP Group must design the overall MIS and calculate resources needed (activity 100-300); and the MIS Group must continually interact with the other two groups on policy matters and also represent the user's needs (activity 100-200). The completion of these three activities results in an MIS plan to be submitted to the Rector (activity 300-500). This constitutes the completion of the project. But the three activities are stated in very general terms. They each are a set of activities. These detailed activities are shown in other figures. In Figure 4 are shown the activities of the Data Team corresponding to the activity 100-400; and in Figure 8 are shown activities of the ADP Group corresponding to activity 100-300. The MIS Committee has no activities of its own other than working with the other two

Figure 4

Work of Data Team and its main interrelationships


Select input and output
130
140
Approve coding structure
150
Review DED
160
MIS Committee
ADP Group
Design DED form
240
Determine coding structure
235
Complete technical part of DED forms
250
Review DED
260
Check for systems needs
230
ADP Group
Data Team
Check for model needs
324
Check for needs of control
346
Check other DED s for completeness
Complete general part of DED forms
Assemble and have DED typed
Prepare DED 1st ed.
100
Prepare I/O table
310
330
List all input
340
350
360
380
400
Check DED completeness with output
320
Check for completeness with output and input list
345
Check for needs of control
326

groups and the interactions are shown in Figures 4 and 8. These interactions and other activities will now be discussed in greater detail.

3. WORK OF THE DATA TEAM

The main task of the Data Team was the development of a Data Element Dictionary (henceforth referred to as a DED) and shown in Figure 4. The DED is a set of definitions and other necessary information (to be listed later) for every element of data required for automated data processing. Much of this data is not exclusively used for automated data processing and is used for manual processing with or without an automated system. But it is essential to automated data processing and the working of computer programs for generating the desired output. It is also essential for reducing ambiguities, inaccuracies and misinterpretations.

The first step in preparing a DED is to list all the data elements that need to be defined. These data elements are dictated largely as a result of desired outputs. The relationship between the desired output and the required input can be best expressed in a tabular form sometimes referred to as an Input-Output Table. An example of part of such a Table is shown in Figure 5. It has many analytical uses and these will be discussed later. For the moment, an Input-Output Table is prepared as shown in Figure 4 (activity 100-310). This identifies the inputs needed. But this list is not complete. For example, it does not include the input for output needed but not yet anticipated. Such output may be demanded by existing management that have changing environment such as different governmental demands for information. Some clues to such needs can be found in the experiences of other institutions or in the DEDs used by other institutions and other countries. In our case, we used the extensive DED published by the National Centre for Higher Education in the United States(1). We checked each of its 104 data element sheets (for the Student and Course Files) for relevance to our institution (activity 310-330). We then examined automated output of other educational institutions in Portugal (activity 310-320).

We also checked for expected needs of the planning models. The needs will not be fully known till the model is operational but it was developed enough to list most of its data needs.

This list was checked with the DED by NCHEMS and the list of inputs needed by other models including the most complex educational model CAMPUS. This was activity 310-324.

---

1) For the most recent edition, see Goddard et al. (1973), Data Element Dictionary, Boulder, United States: National Center for Higher Educational Management Systems at WICHE.

Figure 5

SAMPLE OF INPUT-OUTPUT TABLE

ELEMENTO Nº.	OUTPUT NOME	ALU G1	ALU G2	ALU G3	ALU G4	ALU G5	ALU G6	ALU G7	ALU G8	ALU G9	OTHER REPORTS
1	NOMBRO DO ALUNO	X		X		X	X	X	X		
2	NOME - COMPLETO					X					
3	NOME - ABREVIADO	X		X		X			X		
4	DATA DE NASCIMENTO										X
5	SEXO										X
6	NATURALIDADE		X								
7	ESTADO CIVIL										
8	MORADA				X						
9	TELEFONE					X					
10	ORIGEM ESCOLAR-TIPO DE INSTITUIÇÃO										X
11	TRANSFERÊNCIA										X
12	OR. ESC.CLASSIFICAÇÃO FINAL OBTIDA										X
13	O.E. - ANO DE OBTENÇÃO DA CLAS. FINAL										X
14	RESIDÊNCIA A DATA DO PEDIDO DE ADMISSÃO										X
15	DATA DE ADMISSÃO										X
16	CLASSIFICAÇÃO DA ADMISSAO										X
17	NOME DO PAI	X									X
18	PROFISSÃO DO PAI		X								X
19	NÍVEL DE INSTRUCÃO DO PAI									X	
20	NOME DA MÃE	X									X
21	PROFIESÃO DA MÃE										X
22	NÍVEL DE INSTRUÇÃO DA MÃE										X
23	APOIO ECONÓMICO - TIPO										X
24	APOIO ECONÓMICO - MONTANTE										X
25	SITUAÇÃO HABITACIONAL DO ALUNO										X
26	OCUPAÇÃO PROFISSIONAL DO ALUNO										X
27	CURSO - CÓDIGO	X	X	X			X	X			
28	ANO LECTIVO DE INSCRIÇÃO	X	X	X				X	X		
28A	SITUAÇÃO MILITAR		X								
29	SEMESTRE LECTIVO DE INSCRIÇÃO	X	X	X				X			
30	TURMA EM QUE ESTÁ INSCRITO										
31	SEMESTRE CURRICULAR		X								X
32	SITUAÇÃO INICIAL NO SEMESTRE CURRICULAR										X
33	DISCIPLINA EM QUE SE INSCREVEU	X	X	X				X			X
34	SITUAÇÃO INICIAL NA DISCIPLINA	X									X
35	CLASSIFICAÇÃO OBTIDA NA DISCIPLINA		X								X
36	DATA DA OBTENÇÃO DA CLASS, MA DISCIPLINA		X								X
37	SITUAÇÃO FINAL MA DISCIPLINA										X
38	SITUAÇÃO FINAL NO SEMESTRE CURRICULAR							X			X
39	SITUAÇÃO FINAL NO CURSO										X
40	CLASSIFICAÇÃO FINAL NO CURSO										X

There are other data needs that will never appear in an input-output matrix even if it is complete. This is because some data is needed for intermediate output, or for checking which does not appear as final output. There are many examples of this in Accounting and Financial Operations.

An example would be a subtotal from a cash register, with the sum of all subtotals being compared with the grand total for purposes of control. The subtotals do not appear in any output and yet they are data elements that must be defined. Such data elements are first identified in activity 310-326.

Finally, there is data needed by the data processing personnel for purposes of checking and control. An example is a check digit(1) used along with a student or personnel identification number to ensure that the number is correctly entered. This again is not output related but is an input. Such inputs are identified in activity 310-230 by the ADP Group.

We have now all the inputs required by output, models, control and data processing (330-340). This list must be submitted, while being compiled, to the MIS Committee for their approval. Since the input is so closely related to the output they both are analysed and selected (activity 130-140).

During input selection, each data element must be subjected to close scrutiny and questions like: is it legal (to collect such data)? Is it measurable, meaningful and collectable? Can it be efficiently maintained? And finally, does it have a positive benefit cost ratio?

The redundancy and inefficiency on input and output can often be analysed by using the Input-Output Table. As an illustration, consider the table in Figure 5(2). It is in Portuguese but it can still illustrate the point. Consider data element 30. It has no output. But it may still be needed for control or future use of the model. If so it should be retained.

But often it is there because of an error or the enthusiasm of the analyst or carelessness Or perhaps it was once needed for an output and the output was dropped but the input remained.

Such errors must be checked. This type of error occurred with data element 30 and consequently it was dropped. Or consider data element 9. It is needed by an output but only one output. In this case that data element has to be correlated with other data elements in the data base and hence it needs to remain in the data base. But

---

1) For a discussion, See Hussain, op.cit. pp.157-159.

2) In the last column all remaining reports have been aggregated so as to fit in one page and not have blank rows.

now consider data element 19, also needed by one output. But it does not have to be correlated with other data elements. Why then is it in the data base? Just to generate an output? But that output could perhaps be generated faster and cheaper if done manually.

Keeping the data element in the data base requires that it be coded, keypunched, verified, stored and processed in a computer. This cost of automation may well be much higher than the manual cost especially if the report is seldom generated and in one copy. Such a cost analysis is necessary before inputs and outputs are selected for automation.

In parallel to the activities concerning selection on input and output was the need to design some of the important codes. This was done by the ADP Group (activity 100-235) and approved by the MIS Committee (activity 235-150).

An example of the problems involved appeared in designing the course code. It is shown in Figure 6. This document took no less than 40 man-hours of discussions and arguments. It took very long because it violated the traditional Portuguese concept of the TURMA, a set of sections offered to students at each level in each major. The new code liberated the student from a given set and let him make his own choices given a set of offerings. This raised many questions of educational philosophy, governmental policy, and organisational procedures. It questioned orthodox conventions and in a new university with a progressive management this is possible. Such opportunities must be fully explored and this was done successfully in the case of the Universidade Nova de Lisboa.

Another coding problem related to the structure of the data element "name". Should it be 166 characters that are currently allowed by the National Identification Application Form? Or could it be less? What should be the order of the first, second...and last name? Why not use initials? How should they be ordered? How should they be separated (by blanks or commas)? And could there be two names: one complete name for legal documents and one short name for internal processing? And this structure, must it apply uniformly to all names: student's name, father's name, mother's name, instructor's name and employee's name?

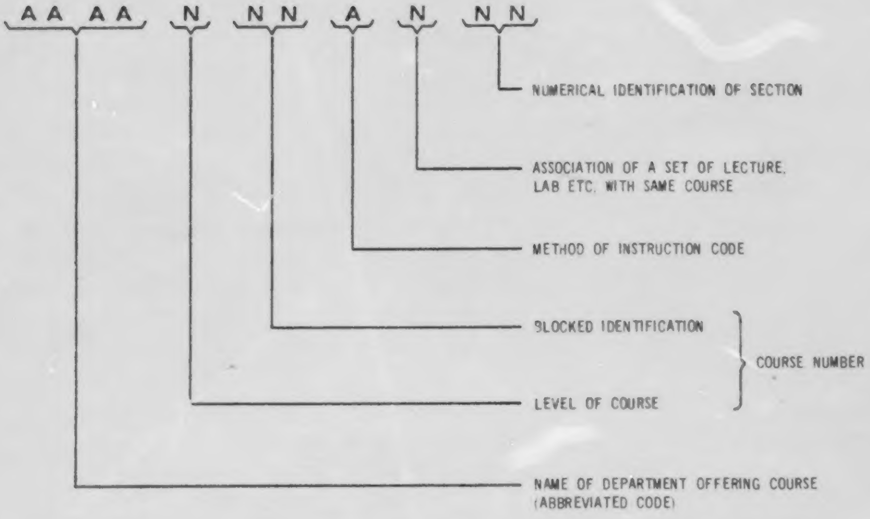
The discussion that followed did not take 40 hours but it was equally heated and much more passionate.

Inputs and outputs being selected, and important codes having been designed, we are now ready to define each data element. The definition (and other information needed on each data element) must be complete and fully specified. To ensure this a DED form was designed by the ADP Group (activity 100-240).

A sample of the DED form is shown in Figure 7. It has two parts. One is on general information completed by the Data Team (340-350);

Figure 5

Structure of Section Code


AA AA
N
NN
A
N
NN
NUMERICAL IDENTIFICATION OF SECTION
ASSOCIATION OF A SET OF LECTURE, LAB ETC. WITH SAME COURSE
METHOD OF INSTRUCTION CODE
BLOCKED IDENTIFICATION
LEVEL OF COURSE
COURSE NUMBER
NAME OF DEPARTMENT OFFERING COURSE (ABBREVIATED CODE)

Note: A Stands for an alpha character.
N Stands for a numeric character.

and another is the technical part completed by the ADP Group (activity 340-250). The technical part is perhaps of little interest to the manager and hence will not be discussed further. The general part, however, needs some illustrations.

For some data elements, there are subfields such as the data element "name" which has subfields of last name, first name, etc. In contrast, some data elements have no subfield but ~~have~~ subclassifications. An example is marital status. Each of its subclassifications must be defined; again unambiguously and exhaustively.

As an illustration of subclassifications consider again the data element of "marital status". Are "married" and "single" sufficient subclassifications? How about the person legally married but separated and living single? How about the legally unmarried but not living single? And numerous combinations? And the widow and the widower?

Do they each need to be subclassified? If so, then they should each be carefully defined.

Some of the information in the general part is self-explanatory, such as the DED number, date when the form was completed, page number and title. All this is for reference. The definition, however, is very important. It must be unambiguous, specific and complete. As an example consider the data element: number of students enrolled.

Figure 7

SAMPLE OF COMPLETED DED FORM

Nº DO ELEMENTO	ALU 06

UNIVERSIDADE NOVA DE LISBOA

Dicionário de Elementos de Informação

DATA	0 8 0 7 7 4
PAG	1 of 1

TITULO

SEXO

DEFINICÃO

O SEXO DA PESSOA

CÓDIGO OU INSTRUCÕES PARA REGISTO

Há só duas categorias: homens e mulheres.

Os códigos a usar são:

H - Homem

M - Mulher

ORIGEM

Boletim de Matrícula

UTILIZADORES

Quadros pedidos pelo I.N.E.

Informação a preencher pelos serviços técnicos

FORMAT	1 A	VARIABLE NAME	SEXO
RANGE LIKELY	1 and 2	LOCATION	Student File
RANGE ALLOWED	1 and 2		
REQUIRED OR OPTIONAL	Required		
CHARACTER COMPOSITION	Alpha		

Is that not obvious? Does it need a definition? Yes it does, because it does not specify what number. Is it the number of student bodies enrolled or full-time equivalents of students? Also, does it include the students enrolled for remedial courses or evening classes without any university curricula? Furthermore, when is the number counted? At registration, ten days later, 20 days later, or at the end of the semester? Answers to these questions will greatly vary the value of the data element.

The next most important item are the assumptions and rules of computations used in cases of data elements that are derived. Consider the check digit discussed earlier. There are at least eight commonly used methods and an almost indefinite number of other formulas for calculating the check digit. Which one is to be used?

Or consider the average class-size. Should it be calculated to the nearest integer or corrected to two or three significant digits? How is the calculation to be truncated or rounded?

Or consider the different amounts of money calculated for the Business Office. How accurate should those calculations be made and how are the numbers to be calculated, then truncated or rounded?

All these rules must be clearly specified. The source of the original data and its use needs to be specified so that when and if the environment changes, appropriate adjustments can be made.

Once all the forms on data elements are completed, they are checked with the complete list of input and each data element that appears in the output (340-345). The forms are then assembled and typed (a total of 56 DED forms) (350-360). They are then checked by the ADP Group (360-260) and with the MIS Committee (360-160). Once approved, they are distributed within the institution for use and comments (380-400).

This DED is the first edition implying correctly that there will be more editions. This should be anticipated since management and their perceptions change, environment changes and finally there is "learning" that results from actual use and operations.

One last observation on the work of the Data Team. The DED discussed thus far in activities 100-400 are only for the priority files. These activities must be repeated for all remaining files.

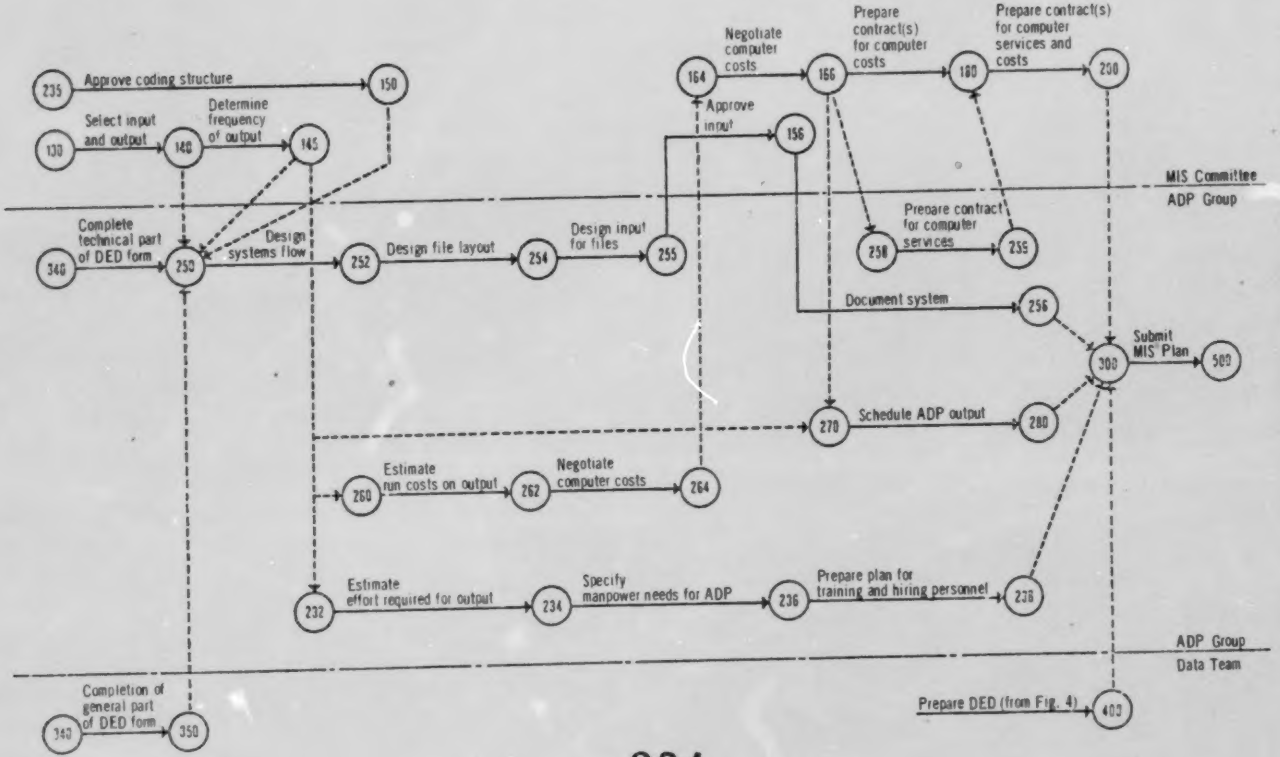
4. WORK OF THE ADP GROUP

Parallel to the work of the Data Team is the work of the ADP Group. Their set of activities are shown in Figure 8 and correspond to activity 100-300 in the overall diagram in Figure 1.

The ADP Group had five distinct sets of tasks. These were:

Figure 8

Work of ADP Group and its main interrelationships


235
Approve coding structure
150
130
Select input and output
140
Determine frequency of output
145
164
Negotiate computer costs
166
Prepare contract(s) for computer costs
180
Prepare contract(s) for computer services and costs
200
Approve input
156
MIS Committee
ADP Group
340
Complete technical part of DED form
250
Design systems flow
252
Design file layout
254
Design input for files
255
258
Prepare contract for computer services
259
Document system
256
300
Submit MIS Plan
500
270
Schedule ADP output
280
260
Estimate run costs on output
262
Negotiate computer costs
264
232
Estimate effort required for output
234
Specify manpower needs for ADP
236
Prepare plan for training and hiring personnel
238
ADP Group
Data Team
Prepare DED (from Fig. 4)
400
340
Completion of general part of DED form
350

1. Estimating and training of manpower
2. Estimating needs of computer equipment
3. Scheduling computer output
4. Designing system
5. Assembling MIS plan

Each one of these tasks will now be discussed in turn.

4.1 Manpower Needs

The main type of manpower needed is for designing and programming the desired output. This can be estimated from the output approved and selected by the MIS Committee (130-140 in Figure 4). This is done by the ADP Group and identified as activity 232-234 in Figure 8. This activity requires information on the frequency of output needed (140-145). It is determined by the MIS Committee and enables the estimation of manpower that will be needed for ADP (Administrative Data Processing). This is identified as 234-236 and the results are shown in Figure 9. These manpower needs are for operational reports only and not for modelling that is required for planning and budgeting. The detailed assumptions and calculations are not shown in this chapter.

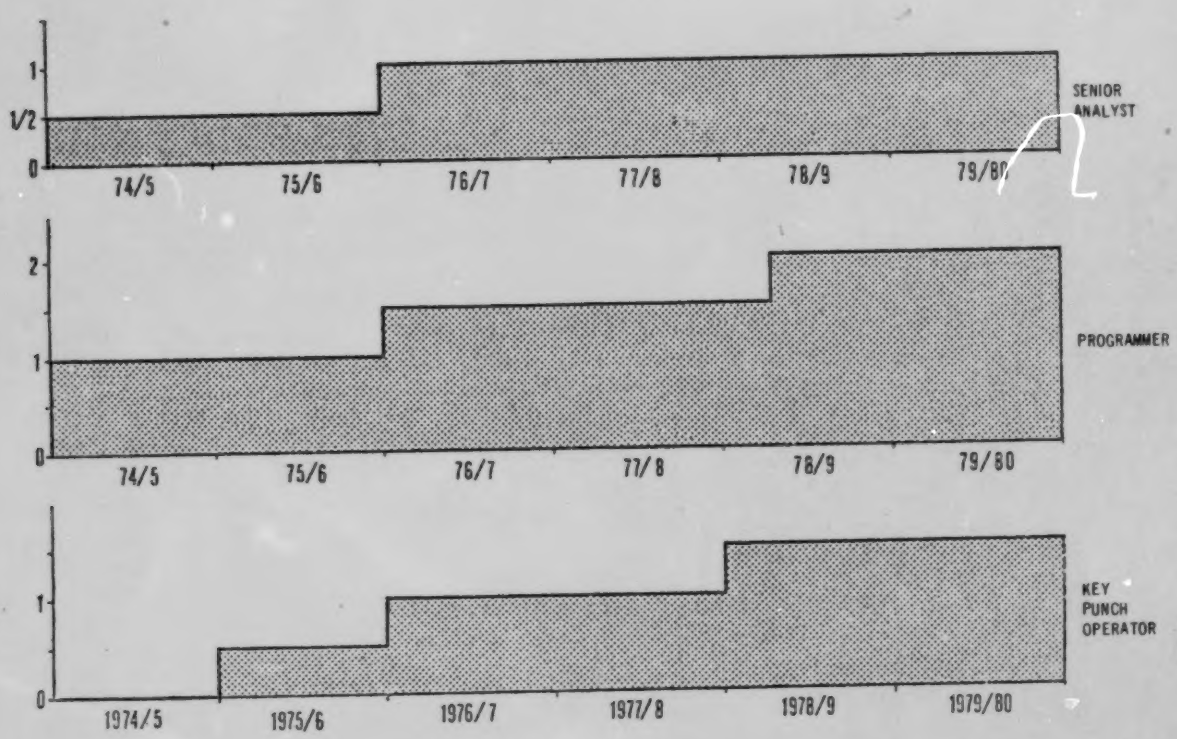
In preparing manpower needs, it was recognised that getting the desired manpower from the market place would be difficult since no degree in Computer Science or ADP is offered in Portugal. Hence the Universidade Nova de Lisboa must have its own capability for self-training its personnel. To facilitate this, a list of books, journals and manuals was prepared (activity 236-238). This is another long lead-time activity and hence this was done early even before preparing the MIS plan. This list is voluminous and does not appear here.

4.2 Equipment Needs

Estimating computer needs and selecting computer configurations to meet these needs is something that is unique to new institutions. Typically, a mature and on-going institution that is big and complex enough to want an MIS system already has a computer for its instructional needs. The administrative needs are then typically marginal costs at most, and sunk costs at best. But having to consider equipment exclusively for educational management is both atypical and very time consuming. And yet this is what had to be done. It also raised another problem. Equipment needs cannot be estimated without a system design, but neither can a good system be designed without knowledge of available equipment. This was the problem of which comes first, the chicken or the egg?

Figure 9

Projected needs of ADP



To overcome this problem, a rough sketch of a desirable system was made assuming a "most likely" set of equipment that would most likely be available. Based on these assumptions, computer costs were estimated (activity 260-262). These were negotiated with the likely supplier of computing services (a nearby university) at two levels: at the technical level by the ADP Group (activity 262-264) and by the MIS Committee (activity 164-166).

Based on negotiations, a draft contract was written in two parts: the technical part by the ADP Group (activity 258-259) and the financial costing part by the MIS Committee (activity 166-180). Put together, there were proposed contracts for different equipment configurations (activity 180-200). Data of these different configurations and the different proposed contracts are voluminous and specific to the Universidade Nova de Lisboa. Hence they are not included in this chapter.

4.3 Scheduling Computer Output

A proposed schedule on the availability of computer output for 1976 was prepared (activity 270-280) based on the frequency of output needed as determined by the MIS Committee (activity 140-145).

4.4 Designing System

The system can be designed after the DED forms are completed (activities 340-250 and 340-350). Much of this work is technical and is not discussed here and therefore the System Flow Chart (activity 250-252), the layouts of the two files (activity 252-255) and the design of file inputs (activity 254-255) are not included. The latter concerns management since they must provide the input through specially designed forms (such as the Admissions Form). Hence the input must be approved by the MIS Committee (activity 255-156). Finally the system is documented (activity 156-256).

4.5 MIS Plan

The MIS Plan is now a matter of assembling and integrating the many documents generated. As a summary, this is shown in Figure 10. This plan along with the DED for the priority files are submitted to the Rector for his action.

5. RESOURCES FOR THE PROJECT

In this section shall be recorded the resources required for completing the project (on preparing the MIS Plan and partial DED). There are two types of resources consumed: time and manpower. Each will be discussed below.

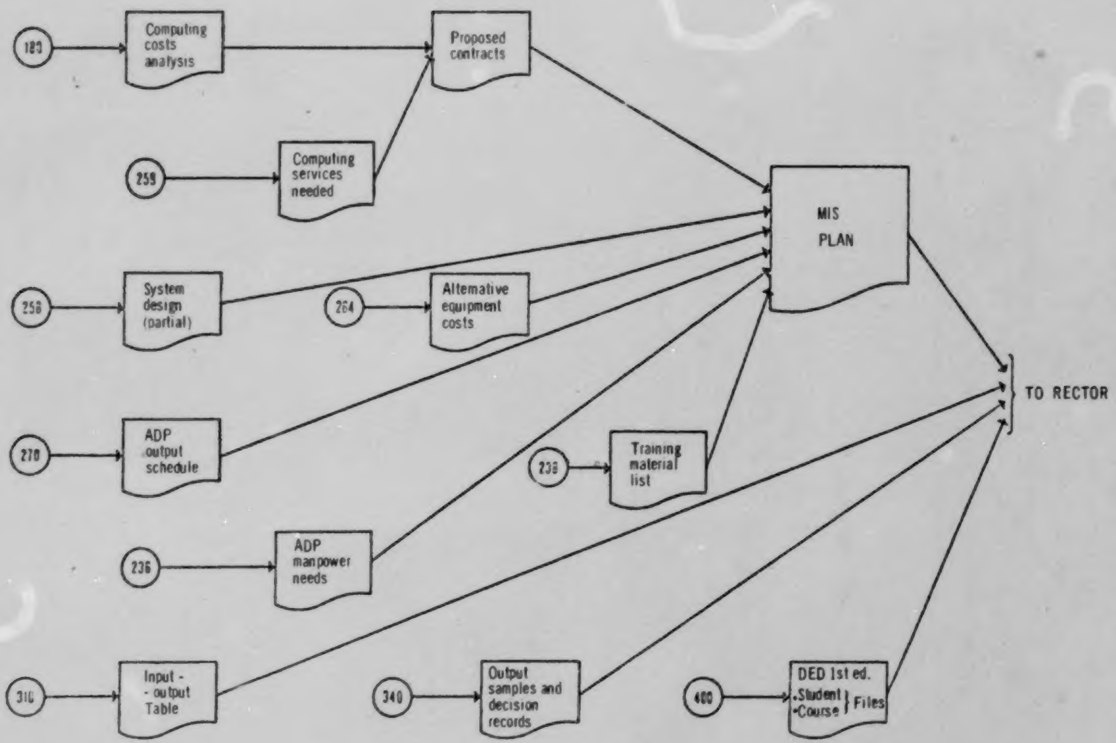


Figure 10

Components of MIS Plan

180
Computing costs analysis
Proposed contracts
259
Computing services needed
MIS PLAN
258
System design (partial)
264
Alternative equipment costs
TO RECTOR
270
ADP output schedule
238
Training material list
236
ADP manpower needs
310
Input - - output Table
340
Output samples and decision records
400
DED 1st ed.
•Student
•Course
Files

5.1 Time Required

The time required for the project can be calculated by assigning times to each of the activities as discussed previously in Figures 1, 4 and 8. It was the diagram used for Project Control and had three time estimates for each activity. Times shown are the actual times. They were - in the detailed network of the project - within the estimated times except for the activities related to computer configurations. The difficult negotiations for computer time were unanticipated and hence the time estimates were underestimated.

5.2 Manpower Resources

The manpower resources used for the project are listed in Figure 11.

Figure 11

MANPOWER USE FOR PROJECT

	Resources involved	Man hours of effort	Lapse time (months)
MIS Committee	2	65	2
ADP Group	3	588	2
Data Team	3	290	1½
Secretarial and drafting	3	224	1½

Note: The groups did not work separately but their time effort is accounted for separately.

6. SUMMARY AND CONCLUSIONS

This final section is a summary interspersed with some conclusions, especially as they relate to new educational institutions. Also, some of the questions raised in the Introduction Section will now be answered more specifically.

1. At the Universidade Nova de Lisboa, the decision to prepare a DED and an MIS plan was somewhat of an accident. The waiting for such an accident is not recommended to other new institutions. Instead the DED and the MIS plan should be prepared deliberately and early in the life of the institution.
2. The DED is essential to any automated data processing but it is also important to manual processing that will never be automated and especially to manual processing that will

later be automated. Its preparation can greatly facilitate the design of forms and procedures needed in the university.

3. An MIS Plan may take years to implement and hence the development of a DED should be independent of the MIS Plan since it relates to MIS as well as non-MIS procedures.

4. A DED sometimes raises questions relating to educational philosophy and organisational procedure. Its preparation could suggest approaches that are innovative though unorthodox and unconventional as was the case of the course code discussed in Figure 5. Such approaches have a better chance of implementation in the early days of the life of an institution. As the institution gets older, the flexibility for innovation is reduced. Hence an early discussion of the DED is very desirable. The resulting dialogue can be very productive in clarifying issues of philosophy, policy and procedures.

5. A DED and an MIS Plan can be prepared with modest resources like seven to ten man-years and within a lapse time of two months. It does not need full-time ADP personnel on its administrative staff or among its teaching personnel as was the case with the Universidade Nova de Lisboa. It utilised a consultant within the country and one from outside.
   What is more important is commitment by top management and support from a capable staff. This was available at the Universidade Nova de Lisboa and was largely responsible for the successful completion of the project.
   Attitude of personnel is not specified as a pre-requisite. Given top management support, the personnel it attracts typically have the necessary attitude.

6. Computer equipment could be a serious problem for a new educational institution, especially if it does not have a sister institution that will offer computer services at marginal or no cost. Paying for ADP services at commercial prices can be very expensive especially for an institution using a planning model. To illustrate the magnitudes involved, consider the average cost of the eight institutions that pilot-implemented RRPM 1-3. It cost them $17,899(1). This cost was per year and calculated at commercial prices. None of them could afford the model at that price and to them it was a service (no cost). These models for planning are very

---

1) For this figure and its detailed breakdown, see Hussain K.M. and J. Martin (1971), "A Resource Requirements Prediction Model (RRPM.1), Report on the Pilot Studies". TR21. Boulder, United States: National Centre for Higher Education at WICHE. pp.16-17.

computer intensive and expensive. At commercial prices they may not be worth it even if afforded. Even computer output for operations may have a benefit cost ratio of less than one, especially for a small and not complex organisation. And this may still prove to be the case with Universidade Nova de Lisboa.

Chapter 12

AUTOMATED ISSUE SYSTEM - BATH UNIVERSITY LIBRARY

by

Ann Chapman and J.H. Lamble
Bath University, United Kingdom

CONTENTS

1. Background 238
2. The Existing Manual System 239
3. The Automated System 240
    3.1 Equipment 240
    3.2 Input Documents 241
    3.3 Transaction Recording 242
    3.4 Further System Details 243
    3.5 System Output 244
    3.6 Back-up Facilities 245

1. BACKGROUND

An on-line automated issue or circulation system was installed in Bath University Library in the Spring of 1975. This system replaced a manual system involving the completion of forms by borrowers and which had been in operation throughout the life of the institution. During the last few years the opportunity had been taken to study and experiment with other methods, none of which had offered any significant improvements or promised long-term solutions to an increasingly complex situation.

Because we are a new university with a new library collection and clearly limited resources to build up an adequate stock we have always been faced with the need to both select our stock with care and judgement and to make sure that this stock is used most effectively. There is a significant relationship between stock acquisition and stock use and it was felt that an on-line automated issue system would provide the management information on the use of our stock that would greatly assist in monitoring our acquisition policies as well as responding most effectively to the complex issuing policy that we have.

The policy of allowing books to be borrowed is central to the idea of the library service given to the teaching and research functions of the university and in meeting the last requirement - the most effective use of stock - the decision was made to categorise the books into different loan periods - 4 weeks for less used books, 7 days for fairly heavily used material and 4 hours for very heavily used books. No distinction is made between categories of readers at this time. This factor is used in helping to achieve the first requirement - good stock selection - in that a lecturer, when recommending a book is asked to state which category of loan period he believes the book should have in addition to the library stock.

It follows from this that, ideally, the issue system we adopt must form an integral part of our stock control processes and provide the information on the use of the books so that we can more easily monitor that use and constantly check our decisions. For example, given a pattern of use and requests for reservations of a particular book, its category of loan period may be varied, either temporarily for short periods of heavy demand or more permanently. That is, our loan system must incorporate a high degree of flexibility.

The management information to be obtained from the issue system is seen as being of as equal importance as the record that must necessarily be kept of the disposition of each item of library stock at any given time.

The specific range of management information that may be obtained from the system is referred to later on. Its requirement led obviously to the need to install an automated system and the choice between an on-line and off-line system followed naturally from an examination of the speed with which some of the information was required.

We had already automated the catalogue of the library stock, and one of the requirements of that development was that a punched card for each book could be produced relatively easily in the event of our automating the issue system.

We had also participated in a project which studied the possibilities and economics of four university libraries sharing a computer installation for the purposes of cataloguing, stock acquisition and stock control.

By the middle of 1973 those particular studies had resulted in a report favourable to such a joint undertaking and the costs and procedures involved to each participant were being considered.

A natural corollary of considering sharing a computer with external bodies was an examination of the possibilities of sharing an installation within our own university and the most likely opportunities would occur with the university administration. This was already well orientated to automation and had made significant progress in introducing data systems on the university's main computer installation which, incidentally, accommodates the library catalogue.

The progress made by the administration was such that they were actively seeking their own installation and it became apparent very quickly that there could be advantages in the library sharing an installation with the administration both in financial as well as system aspects.

Accordingly a specification was drawn up outlining the requirements of the two parties and a number of manufacturers were invited to tender for the supply of machinery against that specification and an indication of the total sum that was available.

After about one year's consultation and negotiation we received four submissions from manufacturers and a decision was made in the summer of 1974 to purchase a Burroughs B1726 configuration.

2. THE EXISTING MANUAL SYSTEM

The manual system was based on a slip filled in by the borrower, checked by the library assistant (who also stamps both slip and item borrowed with a date for return) and then filed until the item was returned. This system was complicated by the three-tier loans policy at Bath as each of the three loan categories used a different variant of the slip system. Thus the short loan collection used a single slip filed in one sequence by author. Seven day loans were recorded on a double slip (not separated for filing) and filed in weekly sequences of date due, within which they were filed in author sequence. The bottom slip was later removed and sent as the first overdue. Ordinary loan books were lent for periods up to a vacation plus a term in length, always being due on the last day of each term. Loans for these books were also recorded on double slips but in this case the slips were separated. The top slip was filed in termly sequences by author and the bottom slip in a single sequence under the borrower's name. This loan period with three return dates per year put quite a strain on the counter staff at the end of each term. There had been pressure from library users to change this period to a four week loan, but this would have increased the circulation routines to an amount that the staff available for this work could not cope with.

During the past couple of years there has been a rapid increase in the number of issues recorded annually. On a loanable bookstock of approximately 66,000 items (excluding periodicals), loans recorded during the period August 1972 to July 1973 totalled over 57,000 while during the period August 1973 to July 1974 the total was 77,153. This number of loans involved a large amount of filing and removing of slips each day, which combined with the illegibility of some borrowers' hand-writing resulted in some filing errors being made. Errors in the files and backlogs of filing, which were liable to build up at any time (e.g. when the number of staff available for such work is lower than usual) both contributed to lessen the up-to-dateness of the files and consequently the efficiency of the service we offered to our users. Other problems being encountered were that of producing overdue notices at the frequency we would like, and that of space in accommodating the trays of slips. This increase in loans also set up a chain reaction of problems in the routines associated with it and an alternative method of recording loans was necessary to prevent a complete breakdown in service.

3. THE AUTOMATED SYSTEM

3.1 Equipment

The automated circulation system at Bath is developed jointly by the University Library and Burroughs Machines Ltd., and shares the use of a B1726 Burroughs computer with the University Administration. The B1726 has an 86K byte store with 14.4 megabytes of head-per-track disc, and a 400 lpm. line printer. Additional equipment for the library comprises a terminal controller, two card/badge reader terminals and a visual display unit.

The card/badge reader terminals are TU100 units through which data is entered into the system. TU100 units will accept a variety of cards and badges from a minimum of 22 columns to a maximum of 80 columns of punched data. These units also have a numeric keyboard, which together with a set of four function keys may be used to input data in non-normal circumstances (e.g. when the book-card is missing). A strip printer on the unit will print responses from the computer at 15 characters per second.

The visual display unit is a TD700 terminal which has a 256 character screen and an alpha-numeric keyboard. This terminal is to be used for enquiries and entering reservation requests, and will have a limited direct data entry facility for the amendments of reader data.

The TC3500 terminal controller will provide the back-up for the system and semi on-line facilities outside normal working hours. It is a small free-standing computer in its own right and in addition to programs for message control and back-up working, can also store lists of blacklisted readers and reservations. In back-up mode, transaction information is recorded on magnetic tape cassettes for transmission to the B1726 when normal processing is resumed.

3.2 Input Documents

The basis of any circulation system is the recording of details of four types of transaction: loans, returns, renewals and reservations. Two sets of data, book and reader details, are required for loans and reservations, whilst only book details are necessary for returns and renewals. At Bath, book data is held on 80 column punched cards which, because of anticipated heavy use, will have plastic edges. The punched data is interpreted visually along one edge of the card. Data consists of (a) a control character identifying the document as a book-card, (b) a unique number (including a check digit) to identify each book, (c) a master number (including a check digit) to link multiple copies for reservation purposes, (d) a category mark or collection mark and shelf mark (truncated classification number), (e) author details - 15 characters, (f) title - 25 characters, and (g) loan category indicator. The University Library catalogue records are already stored on the University ICL System 4 computer and every item added to stock is allocated its own unique number. It has been relatively easy to produce book-cards for the circulation system from a tape of the catalogue records. One problem was that of the small number of columns available for author and title details on the card. Fifteen characters will cope with most personal but few corporate authors, and the 25 characters allowed for the title necessitates truncation and programmed abbreviation of many titles.

The reader's badge is in the form of a plastic laminated badge (the size of a 22 column card) incorporating a photograph, signature and some visual information (reader name, course details, etc.), and has the reader's identify number punched into it. The punched data will allow the badge to be used in the TU100 badge reader terminal, and the punch data, photograph and signature will allow it to be used in a variety of situations at the University where identification is required. There are 10 columns of punched data - a control character identifying the document as a reader badge, followed by a nine digit reader identity number which includes a check digit. The reader's badge is produced at Bath using a Polaroid ID-3 Identification System camera.

3.3 Transaction recording

Three of the four transactions are handled by the TU100 units (i.e. loans, returns, renewals) and the fourth (reservation) by the TD700. A loan is recorded by feeding through the reader's badge followed by up to 10 book-cards. If the reader wishes to borrow more than 10 items at one time, the reader's badge must be reinserted for every further 10 items to be borrowed. To record the return of an item only the book-card needs to be fed through the terminal. The control characters on the book-card and reader badge indicate to the computer a "loan" transaction when the reader character appears first, and a "return" transaction when the book character appears first. The four function keys (which may be used in combination as well as singly) are used before inserting data in special circumstances. The data may either be in card form or using the TU100 keyboard. The special circumstances would be loans with book and/or reader card missing, returns and renewals where the book-card is missing, and occasions when a receipt is required on the return of an item. The strip printer on the TU100 advises the operator that date has passed or failed the validation checks, the fine when an item is overdue and the presence of reservations against items.

Reservations are recorded through the TD700. An identity code is keyed in which calls up a screen format, which the operator fills in with the details of the items to be reserved, and the reader for whom it is to be reserved. Two types of reservation are allowable; either a specific copy is reserved, or any copy of a title may be reserved. The TD700 unit is also used for enquiries. An identity code is keyed as for reservations and the screen format called up is again filled in by the operator. The required information is then displayed on the screen. The four types of query available are (a) what books has a particular reader on loan?, (b) who has a particular book on loan?, (c) what reservations are there on a specific copy, or on a particular title, and (d) when is a particular book due for return?

3.4 Further system details

The reservations facility applies to all loan categories, and in the short loan category allows specification of the period for the reservation so that readers may choose the period most convenient to them. With the seven day and four week loan categories new reservations take their place at the end of existing "queues". Only four week loan items will be recalled by a letter to the borrower, giving him a specific date to return it by and warning him of the fine imposed if not complied with. Recall letters would not be sent in the case of short loan and seven day categories due to the short duration of the loan period, but the records for all categories are marked on the computer file.

In a manual system it can be somewhat difficult to keep a check on those readers whose library privileges have been temporarily suspended. An on-line circulation control system is able to achieve this by simply holding a list of reader identities of readers who are so blacklisted and notifying library staff of this fact when these people attempt to borrow items. A code indicates the reason for blacklisting (e.g. non-return of an inter-library loan, non-payment of a fine) and up to nine reasons may be used (at Bath only seven reasons will be used initially). The reader may be blacklisted and cleared from the blacklist using the TD700.

The computer calculates the date due (and time due in the case of short loan items) and will be programmed to allow for public holidays and start and finish dates of terms which may affect the calculation. In the same way, when fines are calculated, allowance is again made for these factors. Where there is a waiting list of reservations against a book which is normally a four week loan, the computer will treat it as a seven day loan until the waiting list has been cleared.

In any library, books may be removed from the shelves for reasons other than a reader borrowing them. They may still be in the library as in the case of recataloguing and reclassification, or physically away from the library as with items on inter-library loan and at binding. In the manual systems these "loans" have been recorded on slips in the main issue files. In the automated system a set of special reader badges are to be used. They contain no photograph but are printed with an identifier, "BINDING", "MISSING", etc., and instead of a numeric reader number, will have an all alpha code. Some of the report programmes will omit items on loan to these "readers" (e.g. loan time analysis report) while others will include only items loaned to these "readers" (e.g. books at binding report - variant of books on loan report). At present the system will cope with items loaned from Bath to other institutions, but not the other way round, so that a manual file of items obtained for our readers via inter-library loans will still be necessary, at least initially.

3.5 System Output

As well as recording all transaction data, the system is also designed to provide various reports, statistical analyses and letters, either at preset intervals or on request. Much more information on the use of the library stock is available than with a manual system with only one restriction - the amount of storage space available on the computer files to record historical as well as current data. This restriction is primarily an economic one as the storage space available is dependent on the amount of disc space bought.

A 400 lpm printer is to be used to print reports and letters generated by the library system. Three types of letter will be produced using headed, continuous computer stationery. A short text of approximately five lines will be printed by the computer for (a) overdue letters for seven day and four week loans, (b) recall letters for four week loans, and (c) availability letters for reserved items awaiting collection in the seven day and four week loan categories. Following the text will appear the author and title of the book and the date it was, or will be, due for return, or by which it should be collected. Details of the school (and course in the case of students) are kept on the readers file, and will be printed out by the computer as an internal address to which these letters will be delivered via the internal mail system of the University. A few letters will be sent via the external mail and the addresses for these will also be held on the reader files. This will mainly apply to external readers registered as library users. All other reports are printed on ordinary computer stationery. The audit trail is a print of all valid transactions processed the previous day and will be produced daily. The statistical reports are produced at regular intervals. Of the remaining informational reports, some are produced regularly (e.g. books at binding) and others only on request. Reports comprise the following:

1. Audit trail.
2. Loan time analysis. (Statistical)
3. Transaction summary. (Statistical)
4. Overdue books.
5. Reserved books.
6. Reserved books awaiting collection.
7. Books out to special readers (e.g. binding).
8. Books on loan.
9. Books due by a specific date.
10. Reservation history.
11. Books with more than "x" loans in a specified period.
12. Reader listings.
13. Blacklisted reader listings.
14. Fines.

Many of these reports have two or three variants in that they may be restricted to specific loan or reader categories, or to particular periods of time.

3.6 Back-up Facilities

With any automated system, back-up facilities must be provided to cope with the event of failure of part or all of the system. With the system described above, back-up is provided as follows.

1. Failure of B1726 (Main processor)
   Transactions are recorded on the TC3500 on magnetic tape cassettes. List of blacklisted readers and reservations are stored in the TC3500 memory and a reasonable level of back-up is provided. This is also the situation envisaged for weekend and evening working. The magnetic tape recording must then be transmitted to the B1726 on resumption of normal working before on-line processing can again take place. Enquiries and the recording of new reservations can only be done by noting the enquiry/reservation and keeping it aside to process when normal working is resumed.

2. Failure of TC3500
   This affects the TU100 which is attached to the B1726 via the TC3500. It reduces the working situation to one TU100 unit which may have the effect of slowing down the rate of service, depending on the number of persons requiring service at the time.

3. Failure of TU100
   The second unit may still be used, but as with the failure of the TC3500, service may be slowed.

4. Failure of both TU100's
   The following alternatives are possible. Either data could be keyed in via the TC3500 keyboard or a manual data sheet could be used, the information being input to the system later. It is likely that the method used will depend on the individual situation in terms of the length of time the units are expected to be out of action, and the expected workload during that period.

5. Failure of TD700
   Enquiries and reservations would not be possible until the unit had been repaired. Details of reservations could be recorded on cards and input to the automated system when the unit is in operation again.

6. Power failure

This entails a complete cessation of the automated system. A manual system of data entry sheets will be used to cope with this. Data required on the sheet would be as follows:

transaction code,
reader number (loans only),
book number,
master number,
loan category (loans and reservations),
single copy/any copy indicator (reservations only), and
cancellation indicator (reservations indicator).

Renewals would be theoretically possible, but would probably not be allowed in practice as it is not possible, in this situation, to check whether reservations are present against the item being renewed. Reservations would be possible for seven day and four week loan periods, but not the short loan category. Enquiries would not be possible. Depending on the number of persons requiring service, items being returned might be stacked at one side to await the resumption of normal working.

Chapter 13

A COMPUTERISED AND INTEGRATED INFORMATION SYSTEM FOR COURSE MANAGEMENT AT LAVAL UNIVERSITY

by

Marcel Lauzon
Laval University
Quebec - Canada

CONTENTS

1. Introduction 249
2. Description of the system 250
    2.1 Content of files 250
        2.1.1 Course file 250
        2.1.2 Curriculum file 251
        2.1.3 Student file 252
        2.1.4 Other files 253
    2.2 Operation of the system 253
        2.2.1 Course index 253
        2.2.2 Curriculum index 254
        2.2.3 General information 254
        2.2.4 Admission 254
        2.2.5 Registration 257
        2.2.6 Maintenance of educational records 259
        2.2.7 Other uses 259
    2.3 Individual features of the system 259
3. Context of the system 262
    3.1 Objectives 262
    3.2 Background 262
    3.3 Resources and costs 264
    3.4 Flow chart 264
    3.5 Evaluation 264
4. Conclusion 267

Annexes

I. Format of a course description 268
II. Example of the conversational mode 270
III. Format of a curriculum description 271
IV. Example of a curriculum description 274
V. Flow chart of the operations of the Registrar's Office 275
VI. Data Processing Centre Equipment 280
VII. Sample of photocomposition 281

1. INTRODUCTION

For a number of years Laval University has been using its computers for course management, i.e. it has computerised the four major branches of activity involved in course management as understood in the present chapter(1):

- the publication of basic official information on the courses and curricula offered;
- the processing of applications for admission;
- the registration of students;
- the maintenance of educational records.

All the information required for these four major sectors of activity are stored in files which form part of the data bank used by the University for its management purposes.

Information, admission, registration and the maintenance of student records, i.e. the four phases of course management, are closely related. An application for admission presupposes the existence of information on curricula and courses, admission requirements, the duration of studies and the diploma or degree finally awarded. The processing of admission applications must be based on this information as published by the University and it generally culminates in the applicant's admission for the study of a very specific curriculum. Each term the student has to register and choose a number of courses from this curriculum. The courses chosen are recorded in the student's educational record in the student file. At the end of the term the mark (grade) obtained in each of the courses chosen will be entered in the record, an average mark will be worked out for the whole of the record and it will be indicated whether the student is entitled to continue his studies or be awarded his degree, as the case may be. Each student will be given a computer print-out of his record. As information, admission, registration and record maintenance are interconnected, the files on which the management of this sequence is based must themselves be interconnected.

This computerised management education is part of the wider context of the University's administration. For example, a student's registration entails the collection of a fee and therefore has a

---

1) Course management is not used here in its specifically educational sense, i.e. the establishment of a curriculum of courses for a given type of training or the guidance of students in the choice of courses.

bearing on the University's financial administration. Courses are organised on the University's premises and require the presence of academic staff and this creates a link with the accommodation file (space file) and the personnel file. Interconnection between the various files must therefore be more complete than might be required under a more narrow conception of course management.

To meet this scale of management requirements we have therefore set up a computerised system of course management which is itself integrated with the overall management system now being developed.

2. DESCRIPTION OF THE SYSTEM

2.1 Content of Files

To obtain the various computer print-outs required by the University in its course management a mass of information must first be stored in the various files which comprise the data bank. The content of the three files which are most utilised for course management should be at least briefly described, i.e. the course file, the curriculum file and the student file. To obtain certain information it is also necessary to make use of data from the personnel file and the accommodation file.

2.1.1 Course file

A course is defined by the University as all the teaching and study activities which relate to a single subject and cover one term. The course is an elementary unit of teaching and study forming part of a curriculum or several curricula.

The course file contains the following principal elements of information in respect of each course:

- The course number. To avoid duplication the number, which is of the serial type, is automatically printed by the computer;
- The title of the course;
- A summary of the subject of the course;
- The identification code number of the discipline and subdiscipline to which the subject of the course relates;
- The types of teaching used for the course: lecture, laboratory session, seminar, etc.
- Preceding and concomitant courses;
- The number of credits offered by the course. A credit is a unit measuring the amount of work which the University estimates that the course requires; it represents 45 hours spent by the student at the course or in private study;
- The code identifying the faculty and department responsible for the course;

- The access code. This shows which students have access to the course according to the cycle of studies in which they are engaged;
- The season in which the course is normally available;
- The maximum number of students who can be accommodated in one and the same section of the course;
- The total number of students registered for the course in a given term;
- The time and place at which the course is available during the given term.

2.1.2 Curriculum file

The curriculum comprises in the first instance a list of the courses which are compulsory for all students registered for that curriculum. Secondly, it includes the list of optional courses from which the student makes his choice according to the information supplied in the curriculum description. It also comprises a number of credits which the student is expected to acquire in the courses for which he opts even if these credits are not mentioned in the list of compulsory curriculum courses.

The curriculum file contains the following data:

- The serial number identifying the curriculum description and automatically printed by the computer. This number identifies all variants of a given curriculum;
- The title of the curriculum;
- The descriptive code showing the cycle to which the curriculum belongs and the discipline (or areas of studies) concerned;
- The admission code which identifies the secondary school courses which candidates for this curriculum are required to have studied;
- The total number of credits required for the award of a degree. Each course carries a number of credits based on the amount of work the course involves. The student who successfully completes a course gains the credits offered by the course. Each curriculum indicates the total credits required for a degree;
- The number of credits the student is expected to obtain from the optional courses he attends;
- The number of credits available for courses which are entirely left to the students' choice;
- The list of compulsory courses;
- The list of optional courses and information on how to choose them.

2.1.3 Student file

All data concerning the student are entered in this file. The term student applies not only to persons currently studying at the University but also to candidates applying for admission and past students who have already left the University. The principal information to be found in the student file is listed below:

- Student's registration number;
- Date at which his record was started;
- Date of the last entry;
- Student's name and surname;
- Name of father;
- Date of birth;
- Sex;
- Marital status;
- Nationality;
- Mother tongue;
- Permanent address and telephone number;
- Address for correspondence and telephone number;
- Term for which admission application is made;
- Curricula applied for (two choices, by order of preference);
- List of documents still required to complete the application for admission;
- Payment of file registration fee;
- Secondary institution attended;
- Marks obtained at secondary school;
- University's decision as to admission;
- Candidate's reply to the offer of admission to study the curriculum requested;
- Identification of the curricula studies during the University career;
- Term in which the student registered initially for each curriculum;
- Term in which the student completed each curriculum;
- Identification of the student's counsellor;
- Name of the teacher in charge of the student's thesis;
- List of courses chosen at the beginning of each term;
- Fee paid on registration;
- End-of-term mark obtained in each of the courses attended;
- Degrees or diplomas obtained showing term in which awarded;
- Date of leaving the University;
- Reason for leaving.

2.1.4 Other files

The integrated system which Laval University is gradually building up comprises several other files, particularly the accommodation and the personnel files which will be dealt with below. Mention may also be made of the files maintained for the financial branch and the library. All these files already show a satisfactory degree of integration and the system is continually expanding. It may be expected that all files will shortly be integrated into the University's administrative system.

2.2 Operation of the System

The description of the content of the files used for course management may be regarded as the anatomical approach. But what we may call the physiological approach is more interesting for it describes the dynamics of the system. A description of the way in which the system operates and the files are utilised represents a concrete practical approach based on actual experience.

2.2.1 Course index

All the courses which form part of the regular education offered by the University are set forth in two volumes, i.e. the index to first-cycle courses and the index to second and third-cycle courses. The first is a volume of 1,300 pages listing about 5,000 courses and the second contains about 550 pages and 2,000 courses. The courses are all described on the format in Figure 1 and a detailed explanation of this format and the symbols used will be found in Annex I.

Figure 1

FORMAT OF COURSE DESCRIPTION

ARL-10693 GREEK ARCHITECTURE 3cr H

PR: ARL-10690, ARL-10691 LT 2-0-1-6

Study of the patterns and styles of Greek architecture. Building methods. Evolution of patterns, styles and canons of architectural taste. Study of monuments.

39.04 H. Giroux

The programs for processing this file make it possible amongst other operations, to add new courses, make immediate changes in courses already entered in the file, retrieve selected courses and trace the relationships between courses and curricula(1).

---

1) Annex II illustrates how courses are entered in the file.

When an index is updated a print-out of all courses can be obtained at the terminal in a few hours and in the requisite order. The text is then reproduced in offset. Figure 2 represents a flow chart of the production of a course index.

2.2.2 Curriculum index

This index contains a description of all curricula offered in the first cycle, i.e. about 350, with a breakdown by discipline or area of study. This gives a volume of 850 pages which is produced in the same way as the course index by simultaneously processing the course file and curriculum file as shown in Figure 3.

The programs for updating the curriculum file and the course file are the same.

The integrated processing of the two files for the production of the curriculum indexes ensures complete agreement between the two indexes and between all curricula which have courses in common.

Annex III reproduces the format of a curriculum description and Annex IV shows a curriculum description as printed by the computer.

To understand the system it is important to realise that the files which are the source of official information on courses and curricula are also used for the admission and registration of students and the maintenance of student records. This ensures absolute coherence throughout the course management process.

2.2.3 General information

General information for students wishing to apply for admission is published by the University one year in advance. The handbook issued under the title "Admission 75" is an extract from the general information for 1975-76 and shows the curricula offered by the University and the requirements for admission. Although these publications are not yet computerised the information they contain is an exact counterpart of the information stored in the curriculum file.

2.2.4 Admission(1)

The processing of applications for admission comprises a range of operations which are described below and are to a large extent computerised. With nearly 10,000 applications to process computerisation is obviously important.

a) Reception of the application for admission and the necessary supporting documents; codification of the application form; starting of an individual record in the student file on which are entered the

---

1) Annex V shows a flow chart of the operations of the Registrar's office.

Figure 2

PRODUCTION OF A COURSE INDEX

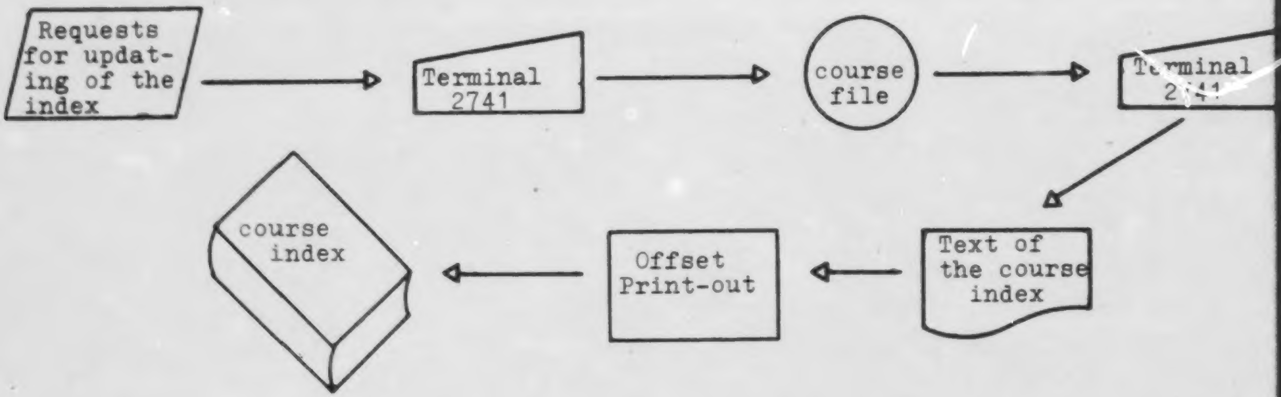
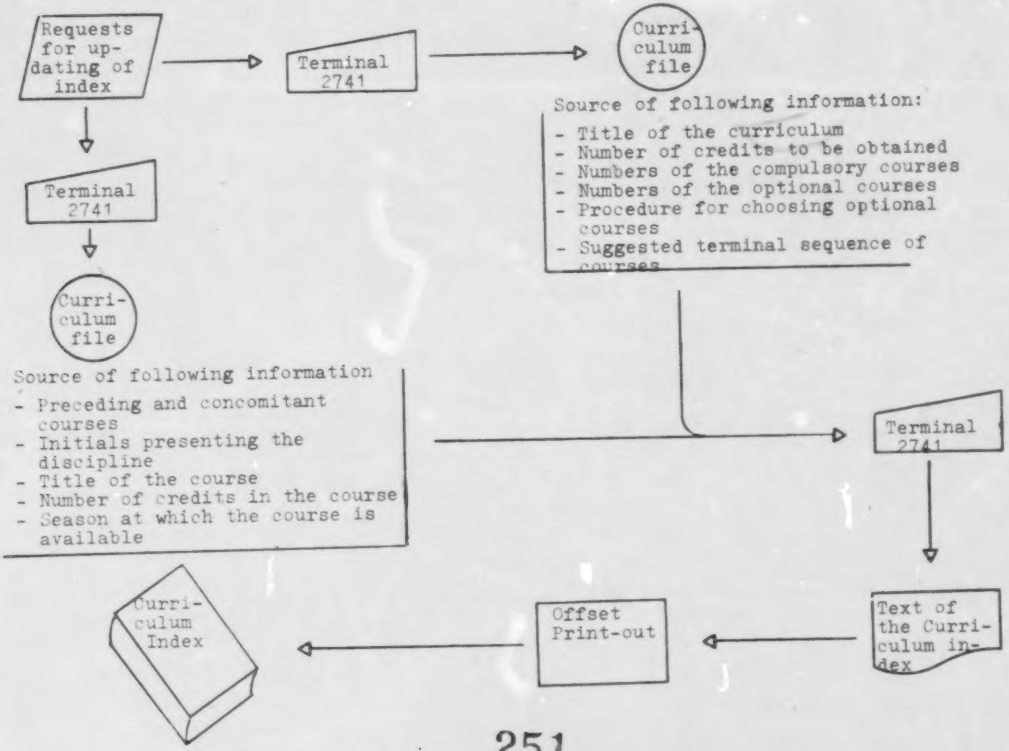

Requests for updating of the index
Terminal 2741
course file
Terminal 2741
Text of the course index
Offset Print-out
course index

Figure 3

PRODUCTION OF A CURRICULUM INDEX


Requests for updating of index
Terminal 2741
Curriculum file
Source of following information:
- Title of the curriculum
- Number of credits to be obtained
- Numbers of the compulsory courses
- Numbers of the optional courses
- Procedure for choosing optional courses
- Suggested terminal sequence of courses
Terminal 2741
Curriculum file
Source of following information
- Preceding and concomitant courses
- Initials presenting the discipline
- Title of the course
- Number of credits in the course
- Season at which the course is available
Terminal 2741
Text of the Curriculum index
Offset Print-out
Curriculum Index

data on the application form, the marks obtained at the secondary school and confirmation that the supporting documents have been submitted. All this is done by teleprocessing.

b) Computer print-outs: acknowledgement of the application, indicating any requisite documents not yet submitted.

- receipt for the registration fee,
- statement of the courses attended by the prospective student while at school as against the courses he will be required to attend in respect of the curriculum for which he has applied,
- form registering the admission board's decision,
- print-out of the lists of applicants for a curriculum who meet criteria laid down by the admission board.

c) Entry of the admission board's decision in the student file.

d) Print-out of the letter informing the candidate of the University's decision.

e) Entry of the candidate's reply in the student file.

f) Periodical print-out of statistics of admission applications: number of candidates, number accepted and number rejected, with reason for rejection. The statistics are compiled for each curriculum, for sets of curricula, for cycles and for the University as a whole. Admission statistics are an essential management tool: they enable demand to be met more effectively by the forecasts they provide of registrations in the ensuing term.

2.2.5 Registration

At Laval University a student wishing to register for a term's study is required to register provisionally the term before (pre-registration). Final registration takes place at the beginning of each term, and the student has a fortnight in which to make any changes in his initial choice of courses. This is a long and complex process involving about 20,000 students who register for hundreds of curricula in which they have chosen close to 100,000 courses offered by over 60 departments. The process is repeated each term. The way in which this computerised system operates will now be shown.

a) Provisional timetable: At the beginning of each term the University departments provisionally allocate premises and compile a timetable for the courses they propose to offer in the next term. This information is entered in the curriculum file. The computer prints the provisional timetable which is then reproduced in offset. A handbook containing the provisional timetable is distributed to students on pre-registration.

b) Pre-registration: This takes place at the end of each term. The student indicates the courses he has chosen on a form, in the light of the provisional timetable. In reality, he need only enter the courses he proposes to attend as the form is already completed in respect of the other data. The courses indicated are entered in his individual record in the student file.

c) *Pre-registration statistics*: Statistics are compiled by courses and sections of courses and submitted to the departments concerned. They are accompanied by a matrix showing how they are related. Each department analyses the situation and changes its timetables where necessary. Changes are entered in the curriculum file.

d) *Final timetable*: When all amendments to the provisional timetable have been registered the final timetable is computerised. It is distributed to students as a handbook at the time of their final registration.

e) *Final registration*: This takes place on the first four days of term. For a student it consists of two stages. He first goes to his curriculum manager who hands him the form showing the courses he chose on pre-registration. He is free to enter any changes he wishes to make on the form. He then takes his form, duly signed, to the registration centre (the main University gymnasium) where he pays a tuition fee depending on the number of credits he has chosen, hands in his form indicating his choice of courses and updates a (computer-printed) card showing his address and other personal information.

As registration proceeds the tuition fees are recorded in the student file together with any changes made in the courses chosen or in the personal information card.

At the peak period in September 1974 over seven thousand registrations were handled in this way on the same day.

f) *Registration statistics*: At the end of each registration day a report is compiled on the number of students per course and per curriculum and is submitted to the officials concerned early the following morning before the registration session begins. Heads of department and curriculum managers then have all the information they require to take any necessary decisions.

g) *Official registration list*: After registration a student may make changes in his choice of courses and these are as usual entered in the student file. At the end of the period allowed for these changes (a fortnight) the computer prints the official lists of students registered for each course and each curriculum. It also produces a financial return for the information of the accounting branch showing any surplus or deficit in the fees collected together with the official statistics on which government subsidies are based. The official lists by courses will be used in compiling students' marks at the end of the term, to ensure that no student is overlooked.

2.2.6 Maintenance of educational records

When describing the content of the student file we stated that all facts concerning the studies pursued at Laval University were registered in this file, i.e. past and present curricula studied (in which term? with what success?), degrees or diplomas obtained. The marks officially awarded at the end of each term are entered in the student file and finally an aggregate return of marks is produced and a copy handed to the student.

This production is one of the most complex in the system. It calls for the use of four files simultaneously, i.e. the student, course, curriculum and personnel files. Programming has to allow not only for a wide range of study regulations but also the administrative decisions taken by curriculum managers. A list of the courses the student has attended must be drawn up and the connection between these courses and the curriculum chosen by the student (compulsory courses, optional courses and courses chosen outside the official range) must be shown. An average figure must then be calculated from the aggregate marks and the consequence of this result on the pursuit of subsequent studies must be recorded. When information has to be produced on this scale the full value of computerisation becomes obvious!

Figure 4 illustrates how the main entries on a student's report are supplied by the files.

2.2.7 Other uses

Other uses of the data bank may be mentioned:

a) Analyses to show the various curricula for which a particular course is used and its relation to other courses, as a preceding or concomitant course. This information is essential in deciding whether a given course should be discontinued.

b) Statistics of attendance at courses each term.

c) Analyses of the contribution made by each department to the training offered by the University, as measured in student credits. This contribution affects the way in which the budget is apportioned.

d) Analysis of the use of accommodation in terms of time and the proportion of places occupied.

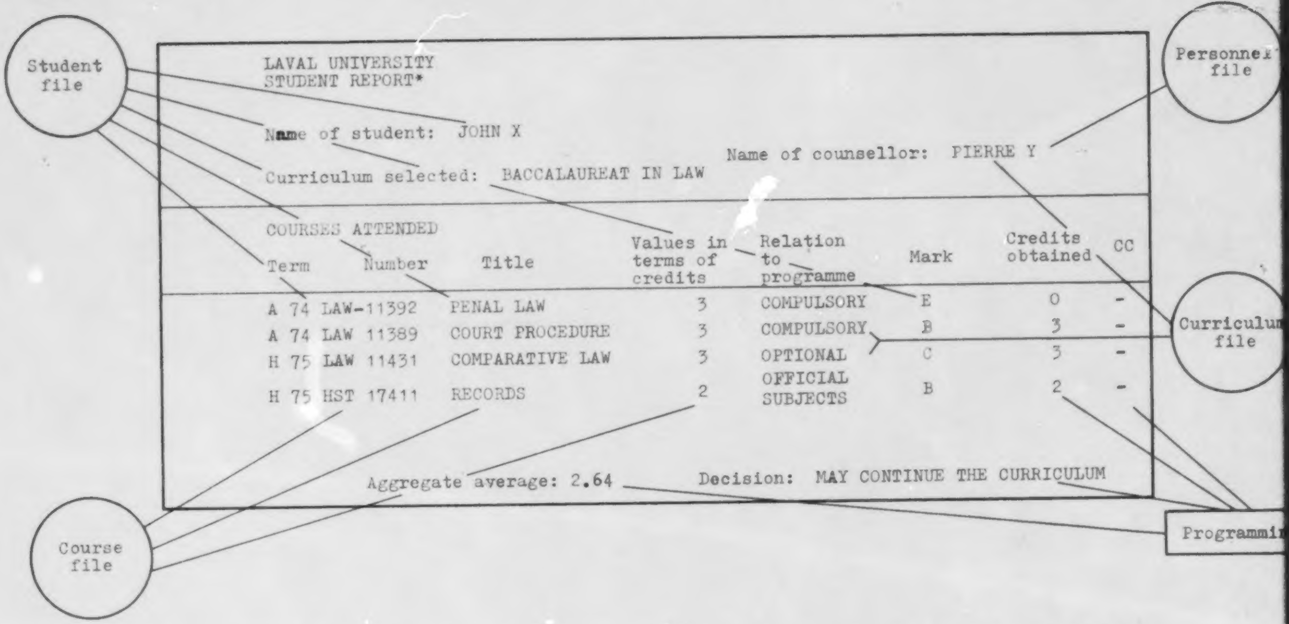
2.3 Individual Features of the System

Certain features which give the system remarkable flexibility are worth a brief description.

a) Flexibility of files: Contrary to the adage: "a place for everything and everything in its place", the files which comprise the present system have no pre-established structure. This means

Figure 4

INTEGRATED PROCESSING OF FOUR FILES TO PRODUCE A STUDENT'S REPORT



LAVAL UNIVERSITY
STUDENT REPORT*

Name of student: JOHN X

Name of counsellor: PIERRE Y

Curriculum selected: BACCALAUREAT IN LAW

COURSES ATTENDED

Term	Number	Title	Values in terms of credits	Relation to programme	Mark	Credits obtained	CC
A 74	LAW-11392	PENAL LAW	3	COMPULSORY	E	0	-
A 74	LAW 11389	COURT PROCEDURE	3	COMPULSORY	B	3	-
H 75	LAW 11431	COMPARATIVE LAW	3	OPTIONAL	C	3	-
H 75	HST 17411	RECORDS	2	OFFICIAL SUBJECTS	B	2	-

Aggregate average: 2.64

Decision: MAY CONTINUE THE CURRICULUM

---

* This is a simplified report form. The actual form contains other details which are not mentioned here. The figure is intended simply to show how the files are integrated.

that space in a file is reserved only for actual data. No position or space is reserved for potential data. The information is simply juxtaposed and no space is left empty. Files therefore vary in length according to the number of elements of information they contain.

Furthermore, the features of each element likely to make up a file are described in what computer scientists call the data dictionary.

The results of these techniques are twofold. In the first place the space they save in the auxiliary memory (magnetic disks) is evaluated at 75 per cent. Secondly they offer the possibility of creating a new variable in a matter of minutes by describing it in the data dictionary. This possibility ensures the continual development of the system on the strength of its file structure.

b) Modular structure: Movement is the essence of life and facts are living things. Information registered yesterday may have to be changed today. One example is the value of a course in credit terms. It is very often essential that the successive values an item acquires should be maintained in the files. To meet management requirement files must therefore reflect changes. Static files would have no more value than historic records and perhaps even less. All data whose variations have to be recorded in the file are registered on modular lines showing the value a particular item had and when. We feel that the flexibility and mobility of the files are an essential factor in management.

c) Teleprocessing: All files are connected to the TELELAVAL system which gives direct access to the data bank they represent. The user may choose a conversational mode for processing whereby he enters the information concerned simply by replying to the computer messages(1). But he may also use a free mode of processing in which he can enter any instructions he wishes provided he conforms to the rules laid down to safeguard data entry. It is the user himself who composes his processing methods. He is free to adopt them, alter them or suppress them. For example there are at present 54 processing methods for the utilisation of student files, 8 of which are used to establish the files and 46 to change them.

The user can also operate from a terminal and order the selection or the print-out of data. In this way he creates a data bank subset which he processes by a sorting program to obtain lists of a definite format either at the terminal or by high-speed printer.

In this way the user makes himself independent of the technicians and paves the way towards "conviviality" as defined by Ivan Illich, i.e. a legitimate choice by which a user has full access to tools

---

1) Annex II gives an example of this method of working.

without having to depend on specialists and without becoming a slave to the tools. This is a very important factor in the use of the system which can be operated by non-specialists or, at peak periods, by auxiliary personnel.

3. CONTEXT OF THE SYSTEM

3.1 Objectives

What is the reason for the system? What benefits may we expect from it? I think I have already largely answered these questions in describing its different uses.

In a few words we wanted to computerise the various levels of management as far as possible to enable the administrators to base their decisions on reliable data.

We were impelled by two facts, i.e. the complex nature of course management and the volume of the data to be processed.

The description of the system gives a good idea of the wide variety of operations involved in course management, i.e. provision of information on curricula, processing of admission applications, registration and maintenance of individual student records. To deal with these matters in greater detail would require an explanation of the curriculum regulations and this would be outside the scope of the present chapter. However, I would draw attention to the administrative impact of the removal of departmental barriers. This has meant that a student is now attached to a curriculum management centre instead of a department. His curriculum may require him to attend courses offered by a variety of departments. Conversely a given course may be attended simultaneously by students working through different curricula. A system of this kind cannot be administered efficiently unless the files are fed from a data bank which accumulates all the marks awarded for a course and enables them to be assigned to each student's record.

The administration of this system is complex and the challenge it presents is all the greater when the volume factor comes into play. When about 20,000 students have to be handled the problem would be almost insuperable without computerisation.

3.2 Background

Historical events rarely have clearcut beginnings. Even though the present management system only began to develop in 1971 the University had already been using computers for various management operations without aiming at an integrated system. The adoption of academic regulations for the whole of the first cycle in January 1971 was a decisive step towards the decision to set up an integrated system.

I do not propose to go into the background history of the way in which the system was built up and the joys and sorrows of those who embarked upon this enterprise. I will merely quote the experience we ourselves acquired and try to deduce the factors or conditions likely to promote success with the hope that, in so doing I may be useful to others.

I will begin by mentioning that there are two sets of academic regulations: the regulations for the first cycle and for those of the second and third cycles. Each entirely covers the cycle it concerns. This overall approach is essential to the operation of the system. What could we do when we had 22 sets of regulations, each faculty or school and even sometimes each department having its own? Where everybody has his own definitions and his own rules for the game no overall system is possible.

A second factor for success was the establishment of a working group with constant contacts between the future users and the computer experts appointed to set up the system. We consider this co-operation essential. It is obvious that the users alone cannot build up the system they need but if the computer experts put the system together on their own it is highly unlikely that the end product will really meet the user's needs.

The third factor of success is the adoption of a critical path. When we began our task in January 1971 we had to meet a deadline, i.e. 1st September 1972 when the regulations for the first cycle adopted by the University council on 12th January 1971 were due to come into effect. We therefore held a number of intensive working sessions at the outset in order to analyse the operation so as to determine its different phases and trace the pattern of inter-dependence linking the wide range of activities involved in the whole operation. Although it was not absolutely essential to evaluate the time required, it did enable us to establish a critical path likely to meet the major deadlines. Without a tool of this kind I personally feel that we would not have completed the task in time.

The fourth factor of success was the flexibility of the information system worked out by the University Data Processing Centre. The elasticity of the files and their modular structure enabled us to operate the system before it was even completed. For example, we had to be able to process applications for admission by January 1972 whereas the initial student reports were not due to come out before January 1973 and the first graduates would not emerge until June 1975. The constantly evolving nature of the system made it possible to stagger the introduction of the scheme over 3 years. I may add that the process is continuing and we are now even remodelling certain sections of the system!

3.3 Resources and Costs

The system was conceived and worked out by Laval University personnel using the University's equipment. Although the cost of the contribution made by the University Data Processing Centre (staff, equipment and material) can be estimated fairly reliably it is difficult to evaluate the contribution from other sources outside the Centre. For example the contribution made by the Registrar's office over three years may be estimated at three persons working half-time. The costs I am putting forward are therefore purely tentative figures.

It must also be borne in mind that although the system was developed initially to deal with course management it was the constant intention of the planners to make it very general so that it could meet the University's other requirements. From this angle, it is a form of investment which the University utilises either for its own purposes or to execute orders for outside customers. This is a fact which cannot be overlooked in any estimate of the cost of the system.

Having made these reservations, I now quote the figures. Programming costs are estimated at $100,000 and testing costs amount to about the same figure.

Running costs are likely to be about $400,000 per year. This represents an annual cost to the university of about $20 per student.

This expenditure can only be judged by considering the utility of the system for the administration of the University, even if it is difficult to evaluate this utility in dollars.

Efficient management is generally a source of profit and I assume that this also applies to educational institutions even if they are not commercial enterprises.

3.4 Flow Chart

I see no need to describe structures which in any case are always closely dependent on some local context. I should, however, note that the University has set up a body to co-ordinate the system which is headed by the executive vice-rector and on which the other vice-rectors, all the system planners and all major users are represented. The Organisation and Methods Service and the Data Processing Centre also play an essential part in this body. It is an essential forum for discussions between the users and the computer specialists and these contacts were a factor of success in the launching of the system.

3.5 Evaluation

a) Positive Aspects: I feel that the above examples of the way the system is used show the full extent of the benefits administrators can derive from a good information base and I will not dwell on this point further.

I would, however, again emphasise the flexibility of the system and its capacity for adapting to new situations because of the elastic modular design of the files. The Laval system is such an all-purpose tool that it has been used for other applications and purposes, e.g. in the management of a hospital which makes use of the University computers by tele-processing.

The constant adaptability of the system enables it to adjust to the user's requirements and to make good its own deficiencies so that instead of deteriorating with time it steadily improves, owing to the new purposes for which it can be used by the creation of new variables and new types of relationships.

Finally, there is the ease with which the system can be handled. Thanks to its conversational mode for processing it can occasionally be operated by untrained personnel. Indeed, it is a matter of only a few minutes to show an operator how to use a working mode for processing. It is easy to see the full extent of the benefits which can be derived from this facility at peak periods, with the use of auxiliary staff (Annex II gives an example of a conversational mode for processing).

b) Negative Aspects: I will not deal here with the negative aspects which are inseparable from any data processing system, but I refer to the danger of excessive centralisation, the impression that the machine is controlling the users and the paralysis which overtakes everybody in the event of a breakdown.

If I merely consider the system described above I find no negative aspects apart from the imperfections which are a feature of any incomplete system. In the initial stage we had to operate the sections of the system already in use while building up the sections designed to come into operation subsequently. We were forced to adopt temporary solutions in a number of cases and this proved to have disadvantages. The necessary adjustments will be made in time.

c) Prospects: I have already emphasised the constantly evolving nature of the system and its capacity to adapt to new situations. I have also pointed out that we had to start using the system before it was complete and therefore adopt a number of temporary solutions. It is therefore natural that we should now be thinking of improvements and the system readily lends itself to improvements because of its flexibility.

- Photocomposition of curriculum indexes is already under study and we expect to reach this next stage in the near future. Working from a magnetic tape containing all the information for a given index and using an appropriate program,a photocomposition firm is printing the text required for the production of the index in offset.

The particular advantages of photocomposition are:

- The printed text has a better finish. (See Annex VII).
- The index is reduced in volume, with a consequent reduction in leads to printing and postal costs.

- Computerised timetable: Some months ago Laval University set up a committee to study the possibility of compiling a timetable with the help of a computer. Using existing files and with a better knowledge of users' needs it may be possible to begin by compiling a timetable of courses which are common to several curricula. At a second stage the timetable might include the courses relating to a particular curriculum. When it becomes possible to forecast registrations for a given term more effectively the computer will be able to supply accurate print-outs of the percentage occupation of classrooms and accommodation management will gain in efficiency.

- Long-distance applications for admission: In theory direct access makes it possible to imagine a new type of admission service in which a representative of the University might visit secondary schools to meet candidates anxious to study at Laval University. Using a portable console and operating from a distance, he could start records in the student file, register the curricula chosen and supply an immediate answer to certain of the candidates' questions, thereby speeding up the admission process. Above all, he could give a human face to what is only too often considered as an anonymous administrative machine.

- Microforms: Certain computer print-outs take the form of a microfilm or microfiche. This technique is likely to be increasingly used. Its main advantage is to save storage space.

- Personalised description of a curriculum: As from January 1976 we shall be able to hand each student a personalised description of the curriculum he is studying, in order to help him in his choice of courses each term. This description will have two main features: it will allow for any amendments made in the curriculum since the student was admitted and will display the marks he was awarded for the courses he has taken so that he will have an accurate idea of what he still has to do to obtain his degree.

- Curriculum change simulation: It will suffice to inform the computer from the terminal of a student's registration number, the number of the curriculum he wishes to study and the term in which he intends to register for that curriculum in order to obtain at the same terminal, an immediate personalised description of the program showing the courses he has already taken just as if the student were already registered for the curriculum concerned. This type of simulation will certainly prove very useful to students and their counsellors.

4. CONCLUSION

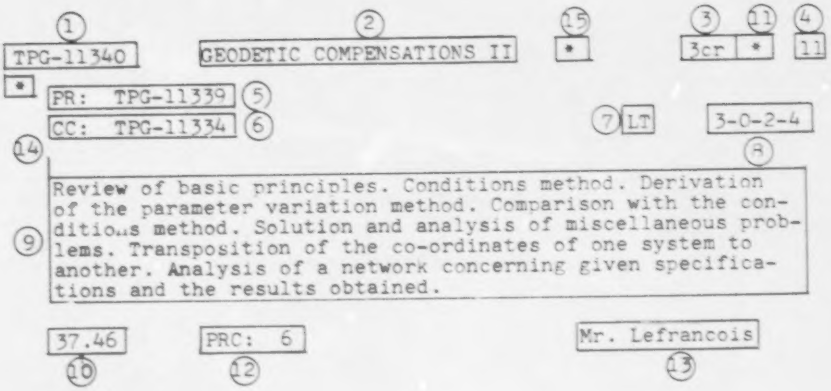
The University uses a computer for the publication of the indexes of the coursos and curricula it offers. Its course management system is based on the information stored in the course and curriculum files and on the information recorded in the student file.

The system comprises the contents of the three files. It is utilised for the production of indexes, the processing of applications for admission, the registration of students and the maintenance of students' record files. Owing to the extensible modular structure of its files the system can be progressively assembled and continually adapted. Because its files are interconnected, it is an integrated system and owing to the ease with which it lends itself to teleprocessing it is highly accessible to users.

This chapter outlines the context of the system, the factors which enable it to be successfully built up and its resources and costs. The study also contains an evaluation of the system and considers prospects of improving it.

Annex I

FORMAT OF A COURSE DESCRIPTION



① TPG-11340 ② GEODETIC COMPENSATIONS II ⑮ * ③ 3cr ⑪ * ④ 11

⑭ * PR: TPG-11339 ⑤

CC: TPG-11334 ⑥ ⑦ LT ⑧ 3-0-2-4

⑨ Review of basic principles. Conditions method. Derivation of the parameter variation method. Comparison with the conditions method. Solution and analysis of miscellaneous problems. Transposition of the co-ordinates of one system to another. Analysis of a network concerning given specifications and the results obtained.

⑩ 37.46 ⑫ PRC: 6 ⑬ Mr. Lefrancois

① Number of the course: the three letters of this alphanumeric code indicate the discipline (or area of study) to which the course relates; see Table T-17.4 in the Annex for the meaning of the initials. The figures identify the course. As long as the course remains the same the figures of the code will remain unchanged.

② Title of the course.

③ Number of credits awarded on the course. For the definition of a credit see the First-cycle Regulations, II-2.6; see also IV-2.2.

④ Season(s) in which the course is generally available; A: Autumn term; H: Winter term; E: Summer term. (RPC, II-1.6, II-4.6 and II-2.78). The timetable published every term will show whether the course is actually available in the season shown.

⑤ Previous courses required for access to the present course. (RPC, II-2.3 and III-7.3).

⑥ Concomitant courses required for access to the present course. (RPC, II-2.3 and III-7.3).

⑦ Types of teaching normally used for this course. (RPC, II-2.2 and II-2.73). The following symbols are used:

A for supporting activities
C for case studies
D for the simulation of concrete situations
G for practical training periods
I for individual tuition
L for formal lectures
M for monographs and projects
P for programmed teaching
R for exercises

S for seminars and conferences

T for practical work sessions

8 Time spent on a course (RPC, II-2.5). The four figures read from left to right:

The first represents hours of attendance per week (PR 1) at formal lectures, conferences, seminars or meetings of all students taking a particular course.

The second represents hours (PR 2) spent on exercises in small groups.

The third represents hours of attendance (PR 3) at practical work sessions or on projects in specialised premises.

The fourth represents hours per week of private work (Pe) estimated by the student.

9 Summary of the content of the course.

10 Faculty and department responsible for the course. (RPC, IV-2.1 and IV-2.2). The figure as a whole shows the faculty, the decimal part representing the department. To interpret this numerical code see Table T-18.3, in the Annex.

11 The asterisk which follows the number of credits awarded on the course shows that the number has been changed since the course was initially inserted into the index. To find out the different credit values see Table T-21.2.

12 Secondary school standard required. This code number relates to Table T-23.1 showing the secondary school curricula which represent the standard the student must already have reached if he is to be likely to pass the course.

13 Name of the teacher responsible for the course.

14 The asterisk under the course initials shows that the course is not part of the curricula designed for students who started their university career in September 1972.

15 This symbol at the end of the title of the course indicates that the title shown in the present index has been changed since it was inserted in the collection of course indexes. Table T-22.2 lists the number of the courses to which this applies.

Annex II

EXAMPLE OF THE CONVERSATIONAL MODE

This operation illustrates how a new course can be inserted from the terminal into the secondary memory, i.e. the course file.

The operation is controlled by the computer which displays a question at the beginning of each line which the operator has to answer. The operator then types his reply to the computer's question. To show the procedure more clearly we have underlined the operator's input.

```
Descriptive code:  1.211. 00BIO-
Title: NERVOUS SYSTEM
Fac. dep.:  30.00
Credits:  1
Season:  A
Teaching method:  LA
Pr-Pe Composition:  1-0-1-1
Previous standard required:  BIO-11134, ANM-11120 or ANM-11121
Concomitants:
Professor: L. Poirier
Code for secondary school:
Description
Course on the anatomical and physiological aspects of the nervous
system, necessary to understand the most frequent neurological
disorders.  End
L
     ok
```

The course is now stored in the computer.

Annex III

FORMAT OF A CURRICULUM DESCRIPTION

1
DESCRIPTIVE CODE: 0.000.00

2
BACCALAUREAT IN ...

DESCRIPTION 0000 3 4 EV : Autumn 1974

Functions: Type ... 5

TOTAL CREDITS TO BE OBTAINED: ... 6

	8	9	10
COURSES AND SEQUENCE 7	Cr	S	T

COMPULSORY COURSES

Number of course	12 Title of course
11	PR: 13 CC:

OPTIONAL COURSES 14 .. credits

credits have to be obtained and the following requirements satisfied:

COURSES FROM WHICH A CHOICE MAY BE MADE 15 .. credits

---

1 Descriptive code. This code shows the cycle to which the particular curriculum and discipline (or area of study) relate.

2 Title of the curriculum.

3 Description number. The description of a curriculum may be changed where appropriate. The variants are identified by a number. The description number entered in the student's record is the number which prevailed when the student was registered to study the curriculum. This subsequently enables the student to claim the credits for the courses he attends in respect of the particular program.

4 Term in which the description shown in the present index came into force. Students who start working through the curriculum with effect from this term are expected to use this description. Those who started previously have to use the description which prevailed at the time they were registered, allowing for any changes which might have been made.

5 The same courses may sometimes enter into the composition of several types of curriculum. For example, a minor curriculum may sometimes be associated with several different types of major curriculum, e.g. type 29, baccalauréat with twinning of major and minor subjects by the student; type 30, general baccalauréat; type 40, certificate.

6 Total credits the student is required by the description to obtain in order to gain his degree or diploma.

7 Courses and sequence. The curriculum description gives the list of courses concerned and suggests a specific sequence for every term throughout the student's university career.

8 Credit value of each course. The value is as shown in course index number 3. It may differ from its previous value. This has no incidence on the record of students who took the course when it had a different credit value; the record continues to show the number of credits the course had at the time the student attended it.

9 Seasons in which the course is normally available:
A; autumn; H: winter; E: summer. The timetable, published each term, shows the courses to be given.

10 Term in which the university organises the course as part of the student's sequence. The term is indicated by a range of figures from 1 to 8. When the term is not identified the figure is replaced by a dash. "As part of his curriculum for a given term the student will normally adopt the courses offered by the University. He will also decide which optional courses and courses of his own choice he wishes to attend in that term" (First-cycle regulations, Section III - 3.22). "The number

of courses a student attends in a given term may, however, exceed or fall short of those laid down in the University curriculum for that term. He should not normally exceed a workload of 18 credits per term." (ibid. III-3.23). A student is required to work for at least 12 credits per term in order to qualify for the status of full-time student.

11 Course number. The essential part of the alphanumeric code is the number; it is the only element which identifies the course. The initials may, where appropriate, be changed but this in no way affects the identify of the course. The initials shown in print-outs of a student's record are always those in force at the time of the print-out.

12 Title of the course. This is automatically the title shown in Course Index No.3. It may differ from the title published in previous indexes. This does not affect the identity of the course provided the latter still bears the same number. On all print-outs of a student's record the title of a course is always the one in force at the time of the print-out.

13 The previous and concomitant courses shown here are automatically those listed in Course Index No.3. In cases where a previous or concomitant course was replaced by another the correction was normally made at the time the course index was compiled but should this correction have been overlooked the number of the previous or concomitant course shown here will probably provide a clue to the new number in conjunction with the tables in the course index.

14 Total number of credits the student has to acquire by attending optional courses as indicated in the curriculum description. An error is always possible in their calculation. The exact total is calculated by subtracting all credits obtainable on compulsory courses plus all credits awarded on courses of the student's own choice from the total curriculum credits.

15 Number of credits freely chosen by the student when deciding the courses he proposes to attend. The student is required to obtain these credits but he is free to choose the courses he wishes to attend from the total of first-cycle courses, subject to the attendance regulations. He may therefore choose any courses listed in the optional group belonging to his curriculum if he so wishes.

Annex IV

EXAMPLE OF A CURRICULUM DESCRIPTION

This is the format for Index No.4 and differs slightly from the format described in Annex III.

INFORMATION
MINOR OR CERTIFICATE
DESCRIPTIVE CODE 1.555.06 NMP 965

Functions: Types 29, 30 and 40 TEV: Autumn 1974
TOTAL CREDITS TO BE OBTAINED: 30

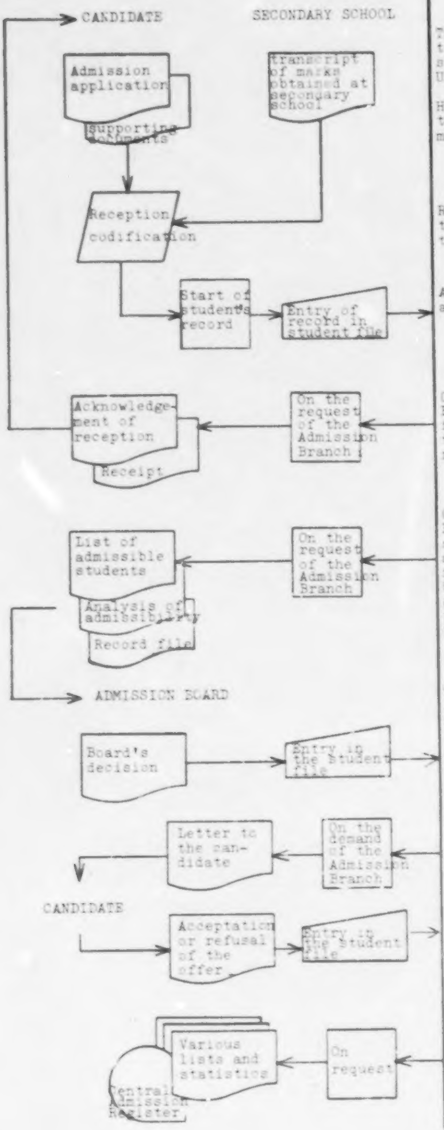
COURSES AND SEQUENCE		Cr	S	T
COMPULSORY COURSES				
IFN-14792	STYLE AND COMMUNICATION	3	AH	1
IFN-14793	SOCIAL PSYCHOLOGY	3	A	2
OPTIONAL COURSES		21 credits		
Students are required to obtain 21 credits and, where appropriate, to satisfy the requirements shown below				
IFN-13610	COMMUNICATION AND SOCIETY	3	H	1
IFN-13597	RIGHTS AND DUTIES IN CONNECTION WITH INFORMATION	3	H	2
IFN-14596	HISTORY OF MASS MEDIA TECHNOLOGY	3	A	2
IFN-13600	ECONOMICS OF COMMUNICATION MEDIA	3	A	3
IFN-13606	COMPARISON OF SOCIAL COMMUNICATION SYSTEMS	3	H	3
IFN-14597	PUBLICITY AND THE MEDIA PR: IFN-14792	3	H	3
IFN-14585	THE PRINTED PRESS PR: IFN-14792	3	H	3
IFN-14586	THE RADIO PR: IFN-14792	3	H	3
IFN-14587	TELEVISION PR: IFN-14792	3	H	3
IFN-14794	HISTORY OF THE QUEBEC PRESS PR: IFN-14596	3	H	3
IFN-14744	COMMUNICATION AND PUBLIC OPINION	3	A	4
IFN-15098	PRACTICAL TRAINING IN JOURNALISM	3	H	4
CREDITS OBTAINABLE ON COURSES FREELY CHOSEN BY THE STUDENT		3		

Annex V

Flow chart of the operations of the Registrar's Office

COMMENTS

PHASE 1 - ADMISSION


CANDIDATE
SECONDARY SCHOOL
Admission application
Supporting documents
transcript of marks obtained at secondary school
Reception codification
Start of student's record
Entry of record in student file
Acknowledgement of reception
Receipt
On the request of the Admission Branch
List of admissible students
Analysis of admissibility
Record file
On the request of the Admission Branch
ADMISSION BOARD
Board's decision
Entry in the student file
Letter to the candidate
On the demand of the Admission Branch
CANDIDATE
Acceptation or refusal of the offer
Entry in the student file
Various lists and statistics
On request
Central Admission Register

The candidate submits his application for admission and the requisite supporting documents to the University.

His secondary school sends in a transcript of the candidate's marks.

Reception of the application by the Admission Branch and codification.

An individual record is started and entered in the student file.

On the request of the Admission Branch, production of the receipt for the fee charged on starting the file and acknowledgement of reception of the application.

On the request of the Admission Branch production of lists of admissible students together with an analysis of admissibility with reference to the courses attended by the candidate at his secondary school, accompanied by the form to be used by the Admission Board to record its decision.

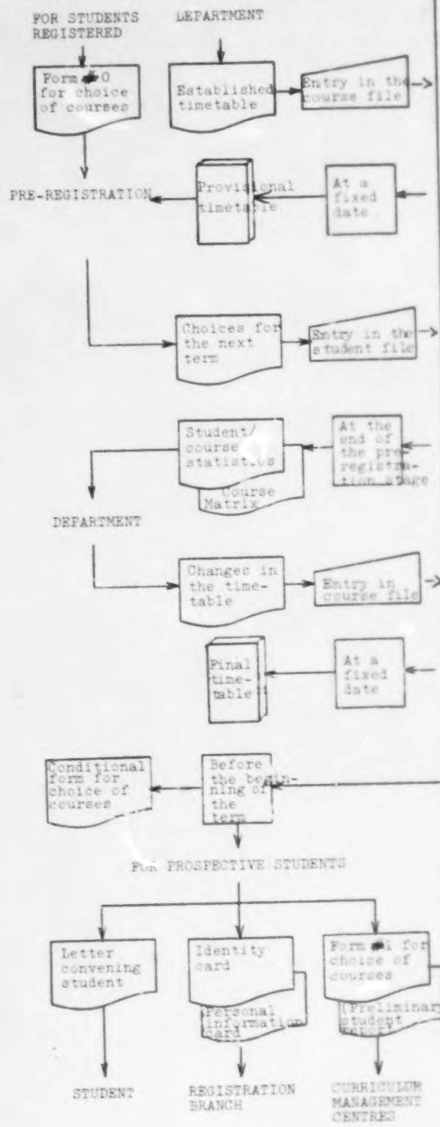
The Board's decision is forwarded to the Admission Branch and is registered in the student file

On the demand of the Administration Branch production of a letter to inform the candidate of the University's decision.

The candidate accepts or refuses the University's offer: his decision is registered in the student file.

On the request of the Admission Branch (or at certain predetermined dates), production of various lists (number of applications per secondary school, per curriculum) and production of statistics (number of students admitted, rejected ...): Copy of the file for the Central Admission Register.

PHASE II - PREPARATION FOR REGISTRATION


FOR STUDENTS REGISTERED
DEPARTMENT
Form #0 for choice of courses
Established timetable
Entry in the course file
PRE-REGISTRATION
Provisional timetable
At a fixed date
Choices for the next term
Entry in the student file
Student/course statistics
Course Matrix
At the end of the pre-registration stage
DEPARTMENT
Changes in the time-table
Entry in course file
Final time-table
At a fixed date
Conditional form for choice of courses
Before the beginning of the term
FOR PROSPECTIVE STUDENTS
Letter convening student
Identity card
Personal information card
Form #1 for choice of courses
(Preliminary student report)
STUDENT
REGISTRATION BRANCH
CURRICULUM MANAGEMENT CENTRES

The Registrar's Office requests the departments to submit a provisional timetable; the data is recorded in the course file.

Production of form #0 for choice of courses designed for all students now registered. It will be used at the pre-registration stage to list the courses chosen.

Production of the provisional timetable.

The student goes to his curriculum management centre and enters the courses he has chosen on form #0. His choice is entered in the student file.

On request by the Registration Branch, production of statistics (by courses/sections/departments) and the matrix of intercourse relationships. These data are sent to the various departments.

After analysis of these data the department changes its timetable, if necessary. The amendments are recorded in the course file.

Production of the final timetable in handbook form.

For candidates who have been admitted conditionally or whose admission is not yet finalised, production of "conditional" forms forwarded to the Admission Branch for the use of candidates.

A few days before registration the files of the candidates definitely admitted are converted into prospective student files.

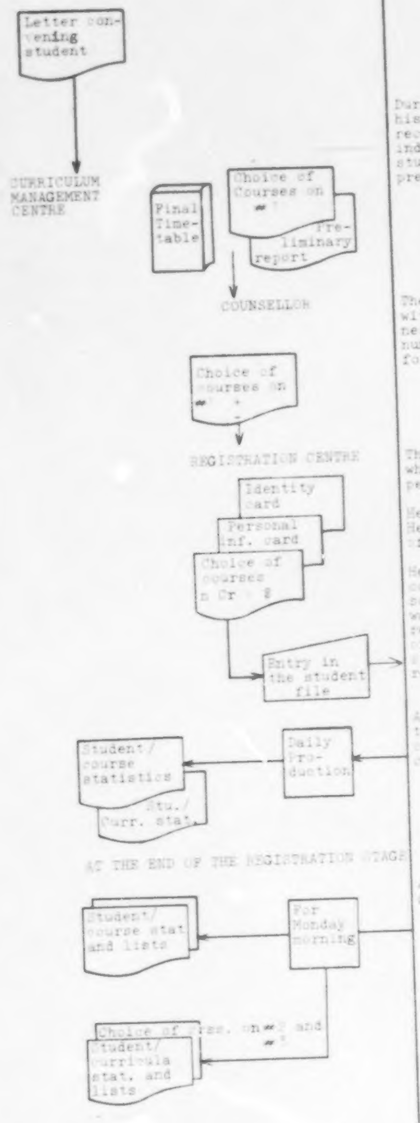
Prospective students are convened by letter.

Production of personal information card for all prospective students and identity card for new students. These documents are forwarded to the Registration Branch.

Forms #1 for choice of courses are sent to the curriculum management centres for all prospective students together with the (preliminary) report on all students already registered in the previous term.

PHASE III - REGISTRATION

COMMENTS



Letter convening student
CURRICULUM MANAGEMENT CENTRE
Final Time-table
Choice of Courses on #1
Pre-liminary report
COUNSELLOR
Choice of courses on #1
REGISTRATION CENTRE
Identity card
Personal inf. card
Choice of courses on Cr. 2
Entry in the student file
Daily Pro-duction
Student/course statistics
Stu./Curr. stat.
AT THE END OF THE REGISTRATION STAGE
For Monday morning
Student/course stat and lists
Choice of crss. on #2 and #3
Student/curricula stat. and lists

During registration week the student goes to his curriculum management centre where he receives his final timetable, form #1 to indicate the courses he has chosen and his student report (if he was registered in the previous term).

The student examines his choice of courses with his tutor and makes any changes he feels necessary. The changes are noted on form number 1. The curriculum manager signs the form.

The student goes to the registration centre where he is given his identity card and his personal information card.

He updates his personal information card. He pays his tuition fees based on the number of credits he has chosen.

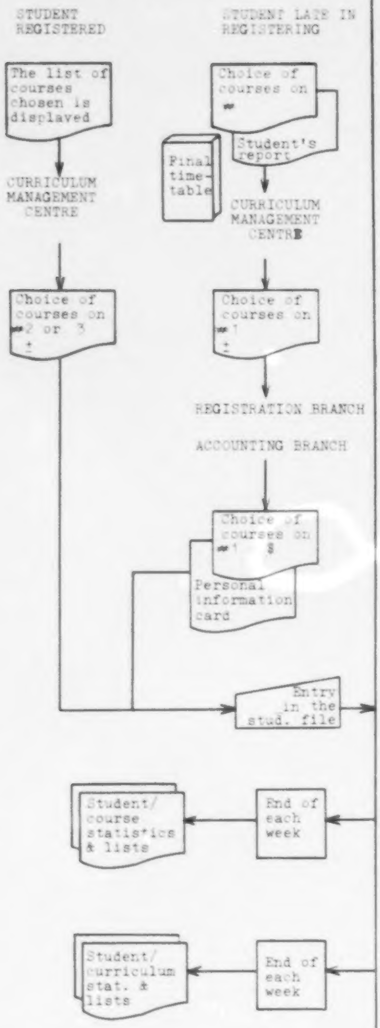
He submits the form showing his choice of courses. The form together with his personal information card are immediately forwarded to the Registrar's Office for the registration of any changes in his choice of courses and his information card in the student file. The tuition fees are also registered.

After each day of registration student statistics are produced by courses and curricula for the use of the departments and curriculum managers.

At the end of registration week, the following data are produced: statistics and lists of students by courses, for the departments.

- courses #2 and #3 chosen
- statistics and lists of students by curricula, for the curriculum managers.

PHASE IV - TWO WEEKS FOR CHANGES


STUDENT REGISTERED
STUDENT LATE IN REGISTERING
The list of courses chosen is displayed
Choice of courses on #
Student's report
Final time-table
CURRICULUM MANAGEMENT CENTRE
CURRICULUM MANAGEMENT CENTRE
Choice of courses on #2 or 3 ±
Choice of courses on #1 ±
REGISTRATION BRANCH
ACCOUNTING BRANCH
Choice of courses on #1 $
Personal information card
Entry in the stud. file
Student/ course statistics & lists
End of each week
Student/ curriculum stat. & lists
End of each week

(Two weeks are alloted for the process described below)

The list of courses chosen by each student is displayed and may be consulted by students registered during the previous week. Any student may apply to change his choice of courses.

A student who arrives late may register in the ensuing fortnight. He follows the same procedure as in the normal registration week.

The student who has registered late makes his choice of courses. The student already registered may change his choice of courses using form #2 or #3.

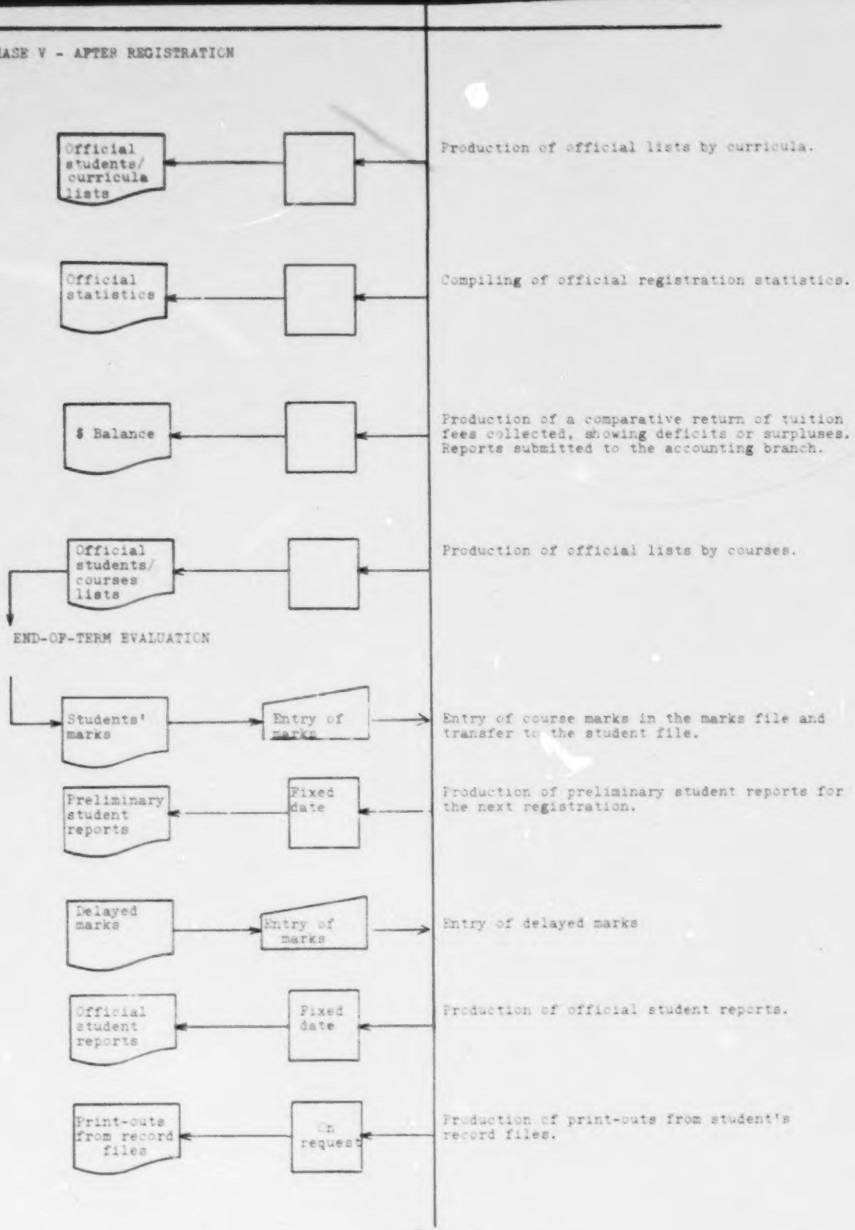
Latecomers pass through the registration branch after which they pay their tuition fees at the accounting branch.

The courses chosen by latecomers and any changes in their choice are registered in the student file.

At the end of each of the 2 weeks allotted for changing the choice of courses, the following data are produced:
- statistics and lists of students by courses

- statistics and lists of students by curricula.

PHASE V - AFTER REGISTRATION


Official students/ curricula lists
Official statistics
$ Balance
Official students/ courses lists
END-OF-TERM EVALUATION
Students' marks
Entry of marks
Preliminary student reports
Fixed date
Delayed marks
Entry of marks
Official student reports
Fixed date
Print-outs from record files
On request

Production of official lists by curricula.

Compiling of official registration statistics.

Production of a comparative return of tuition fees collected, showing deficits or surpluses. Reports submitted to the accounting branch.

Production of official lists by courses.

Entry of course marks in the marks file and transfer to the student file.

Production of preliminary student reports for the next registration.

Entry of delayed marks

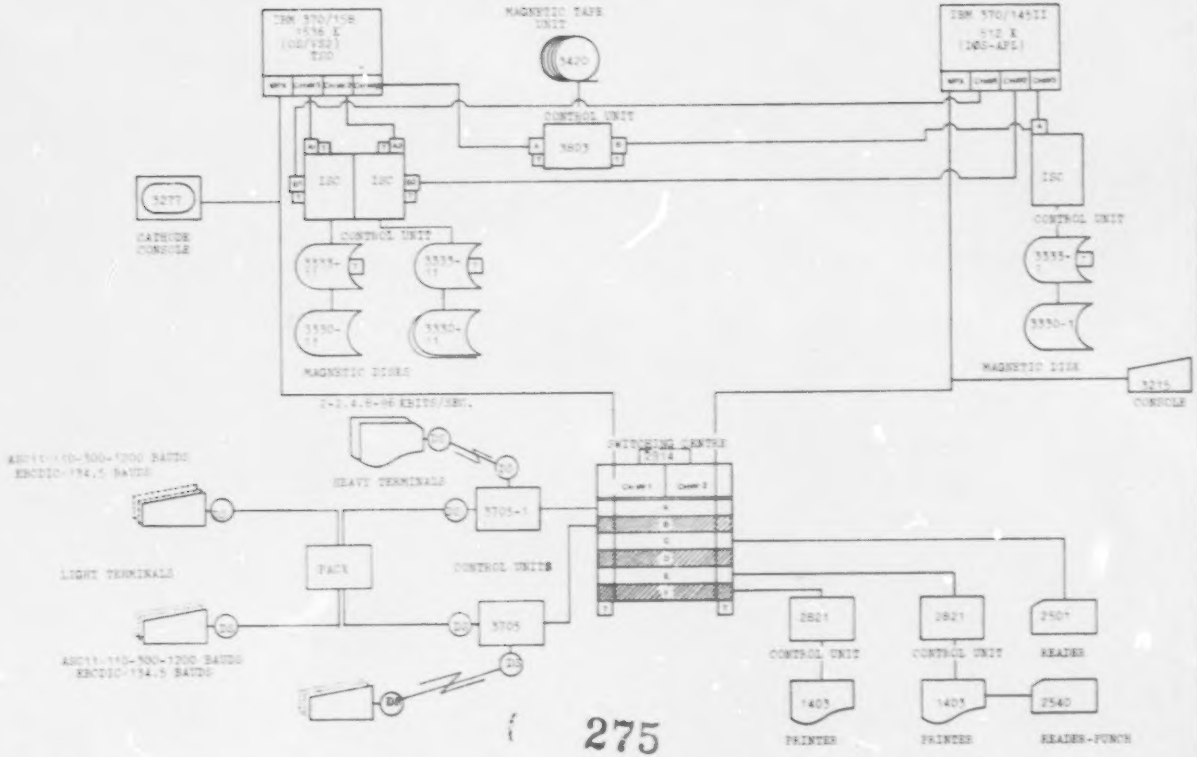
Production of official student reports.

Production of print-outs from student's record files.

Annex VI

DATA PROCESSING CENTRE EQUIPMENT

CONFIGURATION


IBM 370/158
1536 K
(OS/VS2)
TSO
MAGNETIC TAPE UNIT
3420
CONTROL UNIT
3803
IBM 370/145II
512 K
(DOS-APL)
3277
CATHODE CONSOLE
ISC
ISC
CONTROL UNIT
3333-11
3333-11
3330-11
3330-11
MAGNETIC DISKS
ISC
CONTROL UNIT
3333-1
3330-1
MAGNETIC DISK
3215
CONSOLE
2-2.4.8-96 KBITS/SEC.
HEAVY TERMINALS
ASCII-110-300-1200 BAUDS
EBCDIC-134.5 BAUDS
SWITCHING CENTRE
2914
A
B
C
D
E
F
DS
3705-1
PACX
CONTROL UNITS
LIGHT TERMINALS
3705
ASCII-110-300-1200 BAUDS
EBCDIC-134.5 BAUDS
2821
CONTROL UNIT
2821
CONTROL UNIT
2501
READER
1403
PRINTER
1403
PRINTER
2540
READER-PUNCH

Annex VII

SAMPLE OF PHOTOCOMPOSITION

This annex contains a page from the course index reproduced by photocomposition from the course file. It is a sample submitted to the University by a photocomposition firm.

HISTORY

HST-10640 HISTORY OF CANADA II 3cr H
LT 2-1-0-6

Introductory course to the history of Canada since the Rebellion. The subject is studied by topics i.e. cultural (e.g. nationalism), social (e.g. trade unions), economic and political (e.g. political parties, civil liberties).
39.04 R. Jones

HST-10641 ECONOMY AND SOCIETY IN GREECE 3cr A
L 3-0-0-6

Thematic course concerned with certain aspects of Greek thinking at the classical period rather than with economic and social events, e.g. the city and economic problems, attitudes and prejudices connected with work (agriculture, craft industries, trade), interaction of social groups.
39.04 L. Migeotte

HST-10642 ANCIENT HISTORY 3cr H
L 3-0-0-6

Close study of a question or a period selected from (classical) antiquity: Greeee (between 2000 B.C. and 30 B.C.) or Rome (between 1000 B.C. and 500 A.D.)
39.04

HST-10643 SEMINAR IN ANCIENT HISTORY 3cr A
PR: HST-10635 or HST-14325 S 0-3-0-6

Athens at the classical period: attempt at a comprehensive survey. Reading of individual papers followed by group discussion on various aspects of life in the typical democratic Greek city state: politics (operation of the machinery of democracy), society (social groups and problems, family), economy (agriculture, craft industries, trade), war, education, religion (worship of tutelary divinities, popular practices, mysteries), the ideal city as pictured by the philosophers (Plato, Aristotle).
39.04 L. Migeotte

HST-10644 SEMINAR IN ANCIENT HISTORY 3cr A

PR: HST-10635 or HST-14325 S 0-3-0-6

Religious ideas in Rome. Each student reads a paper on the attitude of a particular latin author to religion as evidenced in his works. Group discussions: documentation, method, presentation. The professor provides a bibliography. Authors proposed: Plautus, Varro, Lucretius, Cicero, Caesar, Augustus, Virgil, Horace, Livy, Ovid, Pliny the elder, Pliny the younger, Juvenal, Seneca, Tacitus, Apuleus.

39.04 J. Bernier

HST-10645 SEMINAR IN ANCIENT HISTORY 3cr H

PR: HST-10635 or HST-14325 S 0-3-0-6

Caesar and his times. Each student studies an aspect of Caesar's personality and his work: the man, the politician, the soldier, the reformer, the monarch, the historian, etc. A paper is read, questions are asked and the subject is thrown open to discussion. The professor provides a bibliography.

39.04 J. Bernier

Chapter 14

TOWARDS AN INTEGRATED MANAGEMENT INFORMATION SYSTEM AT THE KATHOLIEKE UNIVERSITEIT TE LEUVEN

by

The Administrative Data Processing Group
Katholieke Universiteit te Leuven, Belgium

CONTENTS

1. Introduction 285
2. Towards an Integrated System (MIS) 285
    2.1 Objectives of the New Organisation 285
    2.2 Stages for Achieving the Objective 285
        2.2.1 The Steering Group 286
        2.2.2 The Project Groups 286
        2.2.3 Activities of Steering and Project Groups . 286
3. Method of Analysis for Information Systems Using a Data Base 287
    3.1 Problem Definition 287
    3.2 Logical Development of this Methodology 288
        3.2.1 First Phase 288
        3.2.2 Second Phase 288
        3.2.3 Third Phase 288
    3.3 How Does this Technique Work? 288
        3.3.1 Problem Definition 288
        3.3.2 Activities Diagram 290
        3.3.3 Information Flowcharts 294
        3.3.4 Comparison of Diagrams 294
    3.4 Use of a Data Dictionary 299
    3.5 Information Requirements 299
    3.6 Establishment or Completion of the Data Base Design 300
        3.6.1 How to Establish or Complete our Data Base?. 300
        3.6.2 Establishment or Completion of the Data Base Design 302

4. A Data Dictionary/Directory 305
4.1 Problem Definition 305
4.2 Principles and Design of the DD/D at the K.U. Leuven 308

Appendix: The Student Information System at the K.U. Leuven 311

1. INTRODUCTION

The Administration at the K.U. Leuven began to be computerised in 1968. Various information systems were developed - such as the Finance System, the Personnel System, the Building System, the Student System as well as other small systems. To a great extent, they were integrated.

In 1971, the K.U. Leuven decided to set up one global system, by means of a Data Base.

The method to set up an integrated system depends on three aspects:

1. The organisation aspect, which will be analysed in section 2.
2. The analysis and methods aspects, which will be considered in section 3.
3. The purely technical aspect, which will be analysed in section 4.

A summary description of the Student Information System is given as an example in the Appendix.

2. TOWARDS AN INTEGRATED SYSTEM (MIS)

2.1 Objectives of the New Organisation: Towards an Integrated Information System (MIS)

All levels of authority from top management to execution must be able to have access at any time to exact, comprehensible, complete and at the same time selected information. In short, information should be adapted to the person receiving it.

2.2 Stages for Achieving the Objective

Detailed activities diagrams and information flowcharts should form the basis of the system. All these flowcharts are prepared by analysts. The method used depends on the structure of each division or subdivision, which in turn depends on the type of activities.

. According to this pragmatic method several activities flowcharts were prepared, but without thereby constituting an organisational chart of the K.U. Leuven.

. All these activities are organised round several information flows; we shall however confine ourselves to those necessary for achieving our objective.

. Through these information flows we shall find all the necessary activities permitting the achievement of part of our objective. We shall achieve the main objectives in the same way as the partial objectives.

. Finally, the charts can be superimposed. The essential information needs will be identified through the overlapping of activities and information (data carriers).

After this preparation phase, we shall become acquainted with the Ad Hoc Committee. This committee consists of a steering group and several project groups. The success of the project depends on the activity of these groups.

2.2.1 The Steering Group

The chairman of this group is a principal director of the administration of the University. The heads of several administrative departments and the manager of the data processing department are also members of the steering group.

2.2.2 The Project Groups

The various units responsible for all administrative activities are subdivided into project groups:

- project groups on "Finance"
"Personnel"
"Extension"
"Building Operation and Maintenance"
"Academic File"
"Building Construction and Conversions"
"Students"

2.2.3 Activities of Steering and Project Groups

1. Activities diagrams and information flowcharts are prepared by analysts in collaboration with the departments.
2. The activities diagrams are checked by the steering group which co-ordinates the preliminary steps for the establishment of the new organisation.
3. The activities diagrams are checked by the various project groups.
4. The steering group sets the overall objectives of the information system and then checks the information flowcharts by means of these objectives.
5. The steering group asks the project group to subdivide the overall objectives into partial objectives and to check the information flowcharts in this perspective.

6. The basic list by means of which the required information will be defined is checked by the steering group.
7. The project groups compile the information needed by means of this basic list.
8. The files are completed by the project groups.
9. A new partial system is designed after the updating of:
   - partial objectives
   - activities diagrams
   - information flowcharts
   - information requirements.

   The absence of overlapping activities in the basic list will also influence this partial system.
10. The steering group develops one global integrated system from various partial systems.
11. In the light of this global system, the steering group specifies which sectors are to be computerised and defines a general computerisation scheme. Thereupon, a list of projects and priorities are submitted to higher authorities.
12. After a decision has been taken by the authorities, the projects can, each in turn, be launched.

3. METHOD OF ANALYSIS FOR INFORMATION SYSTEMS USING A DATA BASE

3.1 Problem Definition

Designing an integrated information system calls for a broad and difficult study. It is, at the same time, a task of organising administrative functions and of developing a project for their automation. The subject we have to deal with comprises operational problems but we shall also be required to answer tactical and strategical questions. These last procedures create new problems for systems analysis, especially because of the links that must be established between the various information units.

In order to define our subject clearly, we shall first indicate the restrictions we shall impose on it:

1. We shall not consider organisational analysis as a whole but we shall confine ourselves to one technique that can be used in systems analysis.
2. We shall not refer to the project documentation but only to a few graphic representations used therein.
3. We are not going to give a technical description of the data base but a possible way to define the concept of a data base,that is, to provide the necessary information and the relationships with data elements.

3.2 Logical Development of this Methodology

3.2.1 First Phase

How shall we prepare a complete diagram of the information system or at least a diagram that can easily be completed? We need a realistic, complete, usable and well-ordered design of the information system to be set up.

3.2.2 Second Phase

How can we effectively incorporate the processed information into the new information system? On the one hand, the data must be collected and stored, and on the other hand the relationships between data elements must be defined. As will be demonstrated, there is a strong relationship between the activities of the information system and the inter-linked data used for these activities. The second phase will consist in developing the first.

3.2.3 Third Phase

A Data Dictionary is needed in order to record in a standardised form the following elements:

- activities
- data elements necessary for these activities
- links between activities
- links between data elements
- links between activities and data elements.

The Data Dictionary must be an active instrument for the design, updating and development of the information system and data bases.

The interaction between the three phases is shown in Figure 1:

- systems analysis
- data base development
- data dictionary.

In order to make the logical development of these three phases possible, we are studying at the K.U. Leuven a standardised technique which can be used in most systems analysis methods. This technique has already been tested on various subsystems and various results have been obtained, some of which will be used as examples.

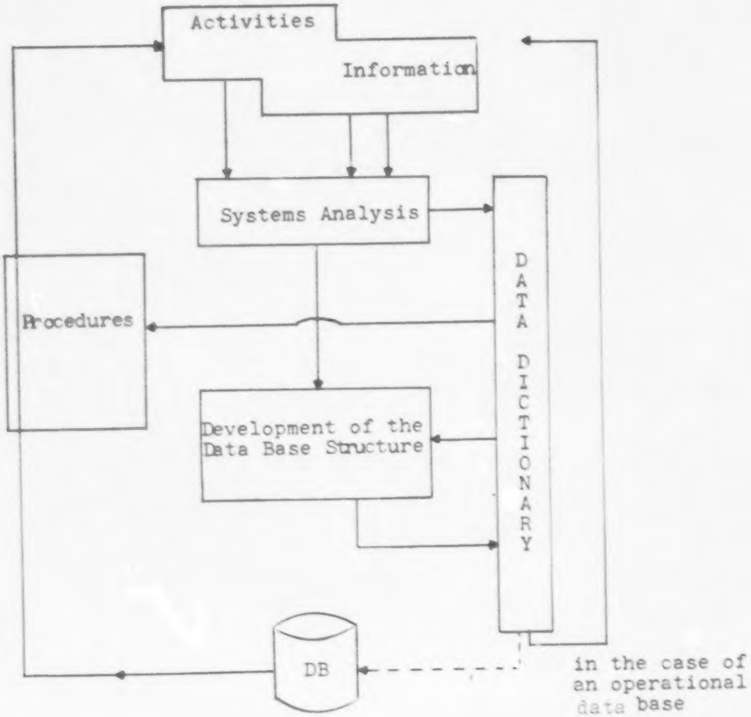
3.3 How Does this Technique Work?

3.3.1 Problem Definition

A method of systems analysis can be developed according to two different principles:

Figure 1

INTERACTION BETWEEN ANALYSIS AND DATA BASE


Activities
Information
Systems Analysis
DATA DICTIONARY
Procedures
Development of the Data Base Structure
DB
in the case of an operational data base

1. **The first principle is based on the structure of the existing system. We examine the existing information system, however rudimentary it may be, and try to adapt it by adding or modifying procedures.**
2. On the other hand, the second principle is based on deduction. We start from a theoretical system, a model that we adapt to the available information and to the objectives of the information system.

Both principles have drawbacks:

- the first limits our possibilities of establishing a sophisticated system;
- the second might provide us with a rather unrealistic data processing system.

However, both also have advantages:

- the first makes it possible to be realistic;
- the second shows us how to introduce the necessary innovations.

This is why we felt that we should use a dual system. We shall represent each of these principles in a diagram and we will have these diagrams interacting until we obtain a new system which meets the requirements we imposed at the beginning of systems analysis.

3.3.2 The Activities Diagram

We took the activities diagram from the method starting with an existing information system.

a) What is an Activities Diagram?

An activities diagram is a diagram in which the different activities of an information system are classified and linked with each other by a hierarchical organisational frame. It is a static diagram which reflects only the structural configuration of the information system. Each activity is subdivided into secondary activities, which in turn are subdivided in other activities until the basic activities or data are found.

b) How is an Activities Diagram Prepared?

The easiest way will be to start from a structures diagram (organisational chart). This structures diagram may very well be the organisational chart of the institution. We draw the organisational chart for the part of the institution for which our information system is to function. This diagram consists of people and departments which are hierarchically linked.

We replace the names of the persons and departments by their corresponding activities and at this moment we observe that some activities are not in their proper place. Because of human factors

Figure 2

IN STAGE 1 WE DRAW THE ORGANISATIONAL CHART OF THE INSTITUTION OR OF THE SUB-SYSTEM OF THE INSTITUTION WITH NAMES, FUNCTIONS AND ACTIVITIES

Activities diagram

Stage 1

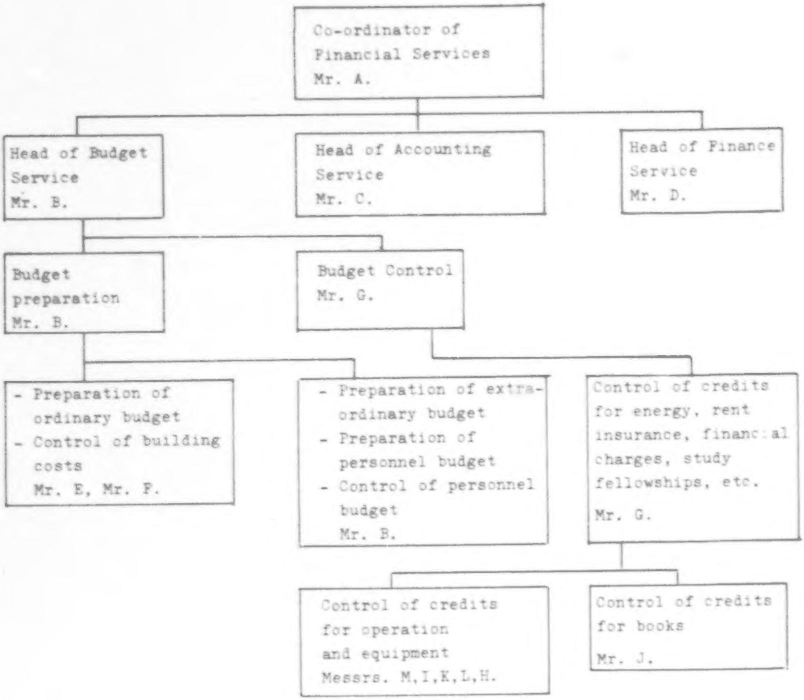

Co-ordinator of Financial Services Mr. A.
Head of Budget Service Mr. B.
Head of Accounting Service Mr. C.
Head of Finance Service Mr. D.
Budget preparation Mr. B.
Budget Control Mr. G.
- Preparation of ordinary budget
- Control of building costs
Mr. E, Mr. F.
- Preparation of extra-ordinary budget
- Preparation of personnel budget
- Control of personnel budget
Mr. B.
Control of credits for energy, rent insurance, financial charges, study fellowships, etc.
Mr. G.
Control of credits for operation and equipment Messrs. M, I, K, L, H.
Control of credits for books Mr. J.

Figure 3

IN STAGE 2 WE SHALL TRY TO INDICATE AS FAR AS POSSIBLE
THE ACTIVITIES OF THE PERSONS IN STAGE 1.

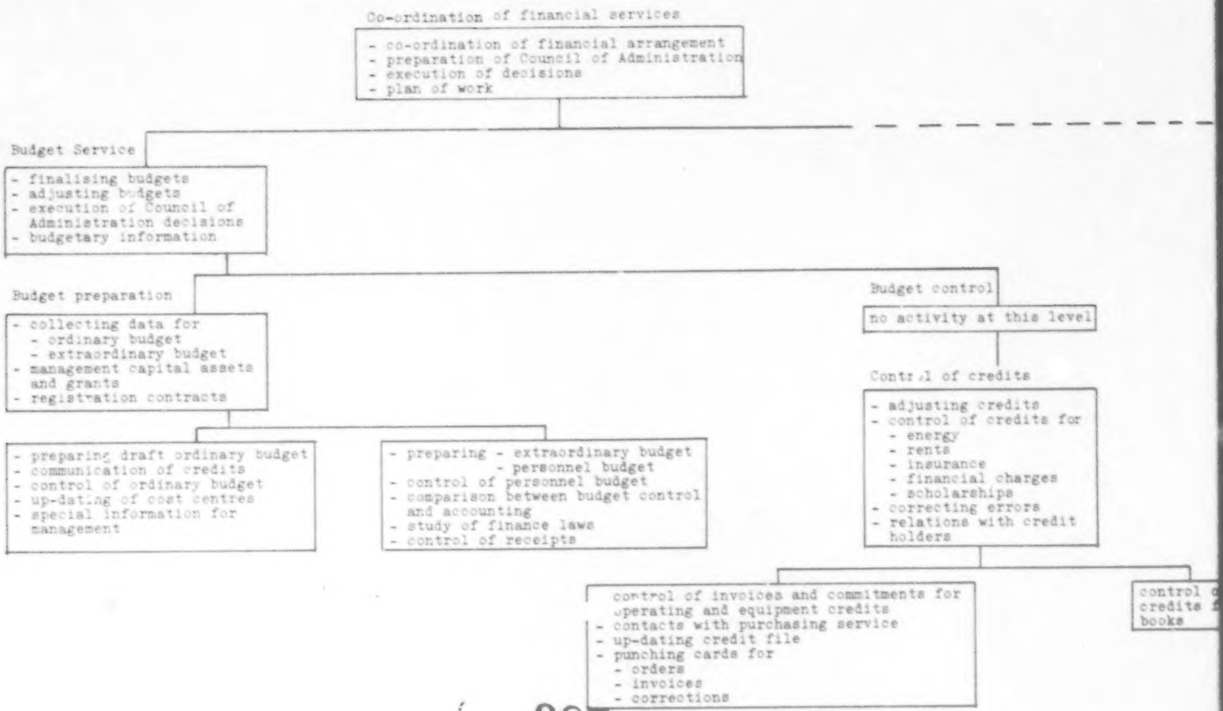
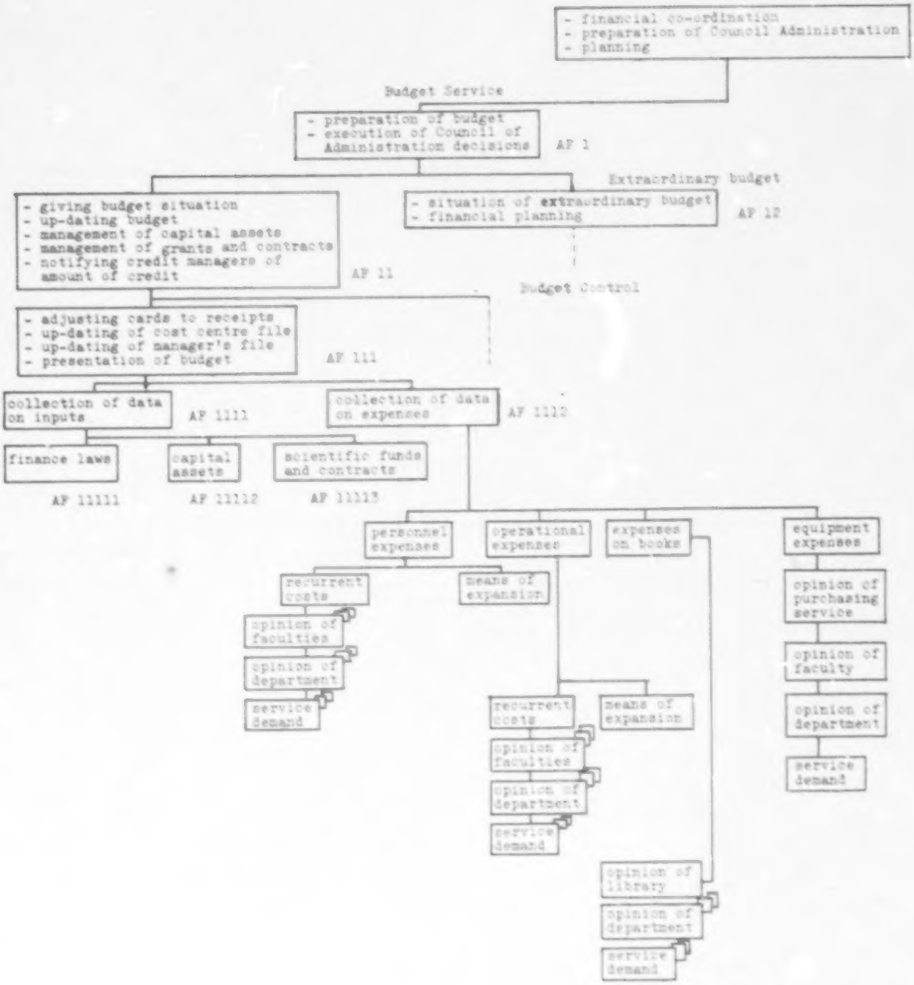

Co-ordination of financial services
- co-ordination of financial arrangement
- preparation of Council of Administration
- execution of decisions
- plan of work
Budget Service
- finalising budgets
- adjusting budgets
- execution of Council of Administration decisions
- budgetary information
Budget preparation
- collecting data for
- ordinary budget
- extraordinary budget
- management capital assets and grants
- registration contracts
- preparing draft ordinary budget
- communication of credits
- control of ordinary budget
- up-dating of cost centres
- special information for management
- preparing - extraordinary budget
- personnel budget
- control of personnel budget
- comparison between budget control and accounting
- study of finance laws
- control of receipts
Budget control
no activity at this level
Control of credits
- adjusting credits
- control of credits for
- energy
- rents
- insurance
- financial charges
- scholarships
- correcting errors
- relations with credit holders
control of invoices and commitments for operating and equipment credits
- contacts with purchasing service
- up-dating credit file
- punching cards for
- orders
- invoices
- corrections
control o
credits f
books

Figure 4

IN STAGE 3 WE HAVE CORRECTED THE ERRORS FROM THE ORGANISATIONAL CHART AND HAVE PUT THE ACTIVITIES IN THEIR LOGICAL PLACE


- financial co-ordination
- preparation of Council Administration
- planning
Budget Service
- preparation of budget
- execution of Council of Administration decisions
AF 1
Extraordinary budget
- giving budget situation
- up-dating budget
- management of capital assets
- management of grants and contracts
- notifying credit managers of amount of credit
AF 11
- situation of extraordinary budget
- financial planning
AF 12
Budget Control
- adjusting cards to receipts
- up-dating of cost centre file
- up-dating of manager's file
- presentation of budget
AF 111
collection of data on inputs
AF 1111
collection of data on expenses
AF 1112
finance laws
capital assets
scientific funds and contracts
AF 11111
AF 11112
AF 11113
personnel expenses
operational expenses
expenses on books
equipment expenses
recurrent costs
means of expansion
opinion of faculties
opinion of department
service demand
opinion of purchasing service
opinion of faculty
opinion of department
service demand
recurrent costs
means of expansion
opinion of faculties
opinion of department
service demand
opinion of library
opinion of department
service demand

and organisational trends some activities, which should normally be linked, are separated, while others are not performed in their logical places or duplicated.

The preparation of an activities diagram for the University Financial subsystem (only the budget part) is given as an example in Figures 2, 3 and 4.

c) Coding the Diagram

A decimal code will be used to identify the elements of the activities diagram (see example AF, AF1, AF11 in Figure 4). The only difficulty encountered is that of structures that are not simple hierarchies (networks). These structures will be identified as if they were only dependent on the first higher level activity.

3.3.3 Information Flowcharts

a) What is an Information Flowchart?

We have outlined the statistical and hierarchical organisation of our activities by means of the activities diagram. On the basis of what we call the information flow, we will now try to develop a system dynamically. The information flow represents activities in the sequence in which they have to take place in order to achieve a specific objective. The activities diagrams, as we are using them, represent an actual situation while the information flow diagrams are used to represent a new administrative organisation, which is designed on a theoretical basis.

b) How to Prepare an Information Flowchart?

The objective is the final goal of the information flowchart. This objective is already known. The easiest procedure is to start from this objective, in order to successively identify the activities. In order to reach the objective, we have to perform activities and performing activities requires resources. In turn, these resources are to be produced by means of activities, and so on.

An example of a university administrative system is given in Figure 5. This information flow is coded in a simple way so that each block can easily be identified.

3.3.4 Comparison of the Diagrams

In order to obtain a realistic but new system, we shall compare the two diagrams and adapt them to each other. For this purpose, we shall use tables as shown in Figure 6.

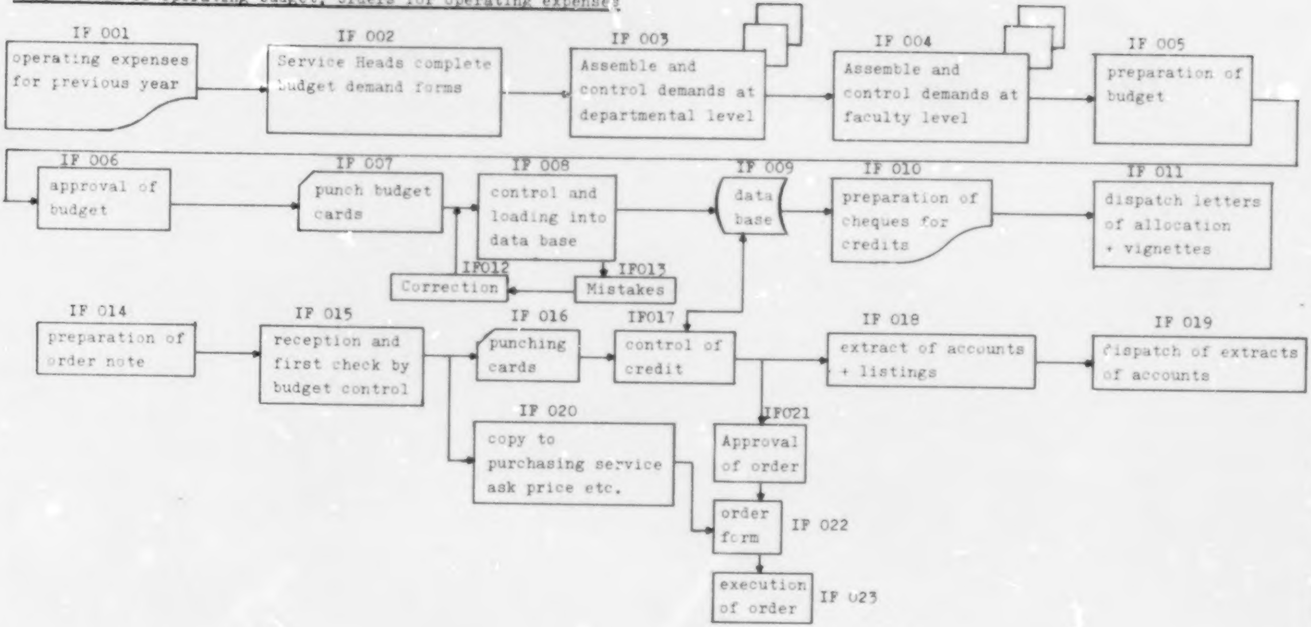
A first category of tables is based on the activities diagram. The different boxes are described in their codification sequence. The corresponding headings of the information flowchart are placed alongside these boxes. We must be careful always to use the same

Figure 5

INFORMATION FLOW CHART

Credit control

Preparation of operating budget, orders for operating expenses


IF 001
operating expenses for previous year
IF 002
Service Heads complete budget demand forms
IF 003
Assemble and control demands at departmental level
IF 004
Assemble and control demands at faculty level
IF 005
preparation of budget
IF 006
approval of budget
IF 007
punch budget cards
IF 008
control and loading into data base
IF 009
data base
IF 010
preparation of cheques for credits
IF 011
dispatch letters of allocation + vignettes
IF012
Correction
IF013
Mistakes
IF 014
preparation of order note
IF 015
reception and first check by budget control
IF 016
punching cards
IF017
control of credit
IF 018
extract of accounts + listings
IF 019
dispatch of extracts of accounts
IF 020
copy to purchasing service ask price etc.
IF021
Approval of order
order form
IF 022
execution of order
IF 023

description for the same activity. The following cases may be encountered:

a) The activity exists in both diagrams.

b) The activity exists in the activities diagram but not in the information flowchart.

Before drawing further conclusions, we must check whether the difference occurs because of some inaccuracy in the description. The unmatched activity will be compared with all the activities available in the column "Information flowchart". The slightest similarity is subject to detailed scrutiny. When we are convinced there is no error in the comparison, we go on analysing this irregularity. We can subdivide it into three groups, of which only one represents a real error.

. One of the groups of activities from the activities diagram with no match-up in the information flowchart does not in fact represent an activity but basic data,such as finance, receipts, expenditures....This observation is entered in the column "Observation". These data elements form the basis of the subsequent development of the data base.

. We can find a second group similar to the first. Here we find all the higher elements of the activities diagram. These are not activities, but groups of management information. This is also important for the structure of the data base which will be discussed further on.

. Finally, the third group provides real errors. These errors can be either activities which should have been in the activities diagram but have been forgotten or it can be redundant activities. These mistakes will be corrected in both diagrams and on the table, preferably in a different colour.

c) The activity does not appear in the activities diagram but appears in the information flowchart. In this case, we must carefully check to see whether the disparity is not a result of a faulty description of the activity. If this is not the case, there may be three explanations:

- The activity in the information flowchart is in the activities diagram of another subsystem.
- The activity is missing from the activities diagram.
- The activity is redundant in the information flowchart.

In these last two cases, the corrections are made as in paragraph b) above. The irregularity of the first type (different subsystem) is noted on the table.

We then proceed to complete the same table in a different manner, starting this time with the information flowchart. We enter in the table all the activities of the information flowchart in their codification sequence. The faults corrected when dealing with

Figure 6

TABLE OF COMPARISON BETWEEN DIAGRAMS (FIGS. 5 AND 7)

No.	Activities diagram		Information flowchart		Observations
	Code	description	Code	description	
1	AF	Finance	IF006	Approval of budget (preparation for Council of Administration)	
2	AF1	Budgets			
3	AF11	Ordinary budget			Management information
4	AF111	Preparation of budget	IF005	Preparation of budget	
5	.				
.	.		IF007	Punching cards	
.	.		.		
.	.		.		
.	.		.		
.			IF011		
.	AF112	Budget control	IF015	Reception and first check of demand sheet	
.	.				
.	.				
.	.		IF016	Punching of cards	
.	.		IF019	Dispatching of extracts	
.	.				
.	.		IF001	Operating expenses for previous year	
.					
.	AF1121	Control of credits for personnel costs			Other flow
.	.				
.	AF112	Control of operating credits	IF017	Control of order credits	
.	.				
.	.		IF018	Extract of accounts	
.			IF021	Approval of order	
.	AF11223	Credits for movable equipment			
.	.				
.	AF11223,30	Faculty of Applied Science			Other flow
.	.		IF014	Preparation of demand sheet	
.	.		IF004		
.	.		IF003	Assemble data	
.	.		IF002		
	AF11223,30,15,03,W02	Credit W02			
	AF11223,30,15,03,W01,1	Old balance			Operational information is in data base
	2	Type of operation			
	21	Order form			IF020,IF022,IF023 other activities diagram
	211	Goods			

the activities diagram no longer appear. Every activity finds its counterpart in the other diagram, with the exception of the basic information and the management information.

There is, however, one error that we have, so far, not been able to detect: the information flow does not represent a logical sequence of events and these events are consequently occurring in a non-linear manner in the system. We must consider whether it is not possible to modify the sequence and introduce more logic in the information flowchart or to change the organisation of the structure of activities; this is very important, as we shall see later.

After having detected and corrected all these errors, we finally have new activities diagrams and information flowcharts. We have indeed taken into account the existing situation as well as the ideal systematic design of the information flowchart.

3.4 Use of a Data Dictionary

At this stage of systems analysis, we already have to call on the Data Dictionary. Have we already incorporated the data found in our diagrams (basic information, activities, management information) in the Data Dictionary or not? What are their relationships to each other and to the new information we have to introduce (if only for testing)? The Data Dictionary will be consulted and completed, if necessary.

A danger in this procedure is that the same data element may be recorded several times under different definitions. Strict standardisation and standards can reduce this risk to a minimum.

3.5 Information Requirements

We do not intend to go any further with the analysis of procedures. We shall rather turn to composition and reciprocal relationships between information elements. We have already identified two types of information:

- basic information (not yet defined)
- management information (not yet defined)

In order to know the content and design of our data base, we must however know all the information necessary right down to the level of the information unit. The following rule can be established:

- to implement an activity (in the information system), we need data elements (information requirements);
- to obtain reliable information, we must implement activities.

By applying this rule, we shall:

- extend the diagrams to the information units
- discover overlooked activities.

For example in Figure 7:

Credit control of operating credit AF 1122, credits for movable equipments AF 11223, credit number W02.

To control this credit, we need the following information elements:

- the previous balance
- the type of operation
- the new balance
- the date.

The previous balance and the date are the information elements. The type of operation includes: the order form, the correction, the adjustment of credit and the entries.

The order form in turn contains new information elements:

- the supplier
- the goods
- the amount
- the taxes (VAT)
- the consignee

The goods, the amount and the taxes are information elements. The supplier implies a new activity: searching for the supplier number. The consignee also implies a search:

- is he a credit holder?
- has he given his agreement?
- is his credit number correct?

The new balance implies an activity:

"previous balance - amount = new balance".

The following observations can be made:

- logical relationships are to be established;
- we work with data which must be obtained by means of the Data Dictionary;
- new elements can be introduced in the Data Dictionary.

At this precise point of our analysis, we have collected all basic material. We know the activities, the information elements and their reciprocal relationships. We have an activities diagram detailed in such a way that it contains few activities but a large number of information elements. The information flowchart shows us the relationships between these information elements and the activities.

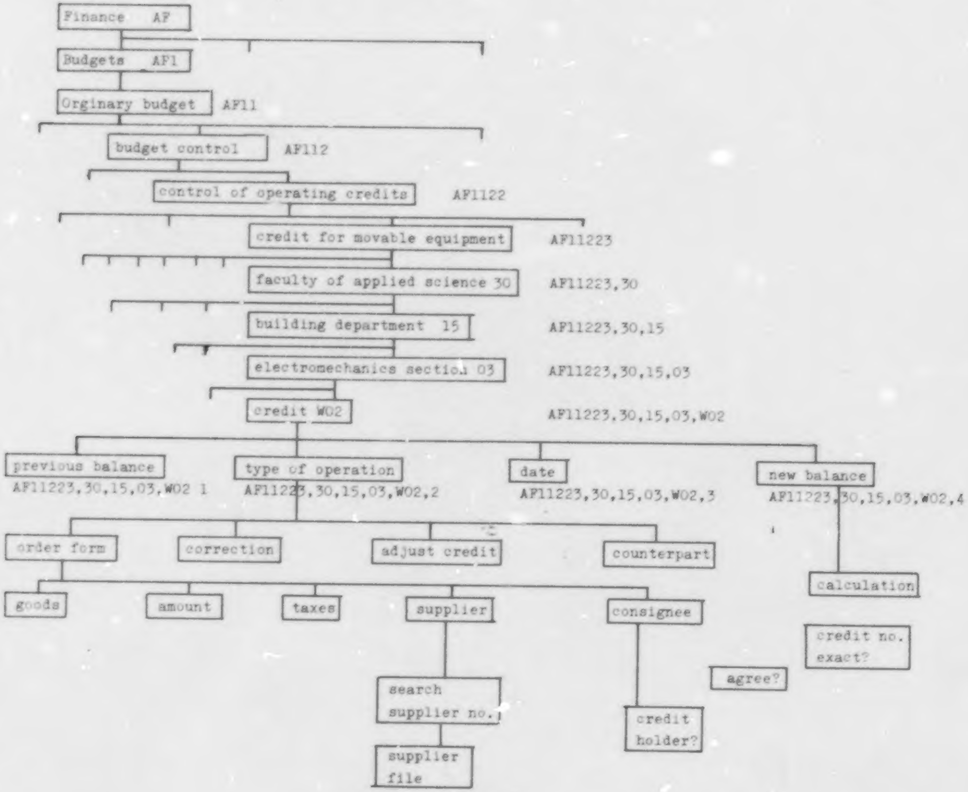
3.6 Establishment or Completion of the Data Base Design

3.6.1 How to Establish or Complete our Data Base?

How are we now going to handle all the information elements in order to develop a data base that can efficiently be used in the information system?

Figure 7

DETAILED ACTIVITIES DIAGRAM


Finance AF
Budgets AF1
Orginary budget AF11
budget control AF112
control of operating credits AF1122
credit for movable equipment AF11223
faculty of applied science 30 AF11223,30
building department 15 AF11223,30,15
electromechanics section 03 AF11223,30,15,03
credit W02 AF11223,30,15,03,W02
previous balance AF11223,30,15,03,W02 1
type of operation AF11223,30,15,03,W02,2
date AF11223,30,15,03,W02,3
new balance AF11223,30,15,03,W02,4
order form
correction
adjust credit
counterpart
calculation
goods
amount
taxes
supplier
consignee
credit no. exact?
agree?
search supplier no.
credit holder?
supplier file

1. We must first check that the relationships we have established are really those that we shall use most frequently. We shall do this by comparing the activities diagram and the information flowchart with the Data Dictionary.

. What is the sequence in which the various information flow branches encounter the activities and information elements they need?

. How many times do we meet the same element in the information flow, or how many times is the same element in the Data Dictionary mentioned as being used by other subsystems?

. Is it necessary, each time there is a basic adjustment, to adjust higher level information elements (management information)?

The answers to these questions may enable us to modify the structure of the activities diagram. At any rate, all these elements must be recorded on the activities diagram since they influence subsequent decisions.

2. Establishment of a logical data base. Our design is MIS-oriented, which of course will have an influence on the method of designing the data base. We shall thus speak of different data bases; logical data bases bear a strong resemblance to the activities diagram and are influenced by the information flowchart.

3. As a physical data base, we shall use the most frequent combinations extracted from the logical data base, which is defined in the activities diagram and from elements mentioned in (1) above.

4. Adjusting the Data Dictionary. By now we have frequently consulted the Data Dictionary and we have incorporated in it elements labelled "Project". We shall now adjust the Data Dictionary to the data base references and change "Project" to "Test".

3.6.2 Establishment or Completion of the Data Base Design

We shall establish or complete the data base in three stages, each time making use of the Data Dictionary. We should, however, point out that this intervention of the Data Dictionary has not yet been tested at the K.U. Leuven. Another group of the Data Processing Department is currently working on the Data Dictionary project.

The data base will be established in three phases:

1. Subdivision of the activities diagram into logical groupings and establishment of links with the existing part of the data base.
2. Comparison of these subdivisions with information flows and fixation of the logical data base.
3. Fixation of the physical data base in the light of the above two points.

1. Subdivision of the Activities Diagram

First, we have to subdivide the detailed activities diagram into different logical parts. For this purpose, certain principles must be observed. The activities diagram must be subdivided in such a way as to obtain logical sets (Figure 8).

A first principle is the subdivision of information into basic information and management information. Thus, in our financial activities diagram, we shall consider all information located at a higher hierarchical level than activities as management information.

What is located below and among activities is generally tactical and operational information and can easily be classified in logical groups, such as defining the outline of the budget, budgetary control, activities, control of the personnel budget, expenditures, etc...

The second principle is to subdivide the tactical and operational information in logical groups forming a whole from the standpoint of structure, implementation and accessibility.

If the management information is too large to form a single group, we must subdivide it in a similar way. Nevertheless, we must always be careful to choose the logical sets in such a way that, together, they will form a new hierarchical structure.

2. Establishment of a Logical Data Base

We begin by establishing a logical data base because we have to know first what we ought to do in order to establish thereupon the physical structures with which we have to work. Had we used the inverse approach, we would have been inclined to establish our procedures and activities in the light of the physical data base with which we have to work. In the first place, we shall also subdivide the information flows into logical groups of activities.

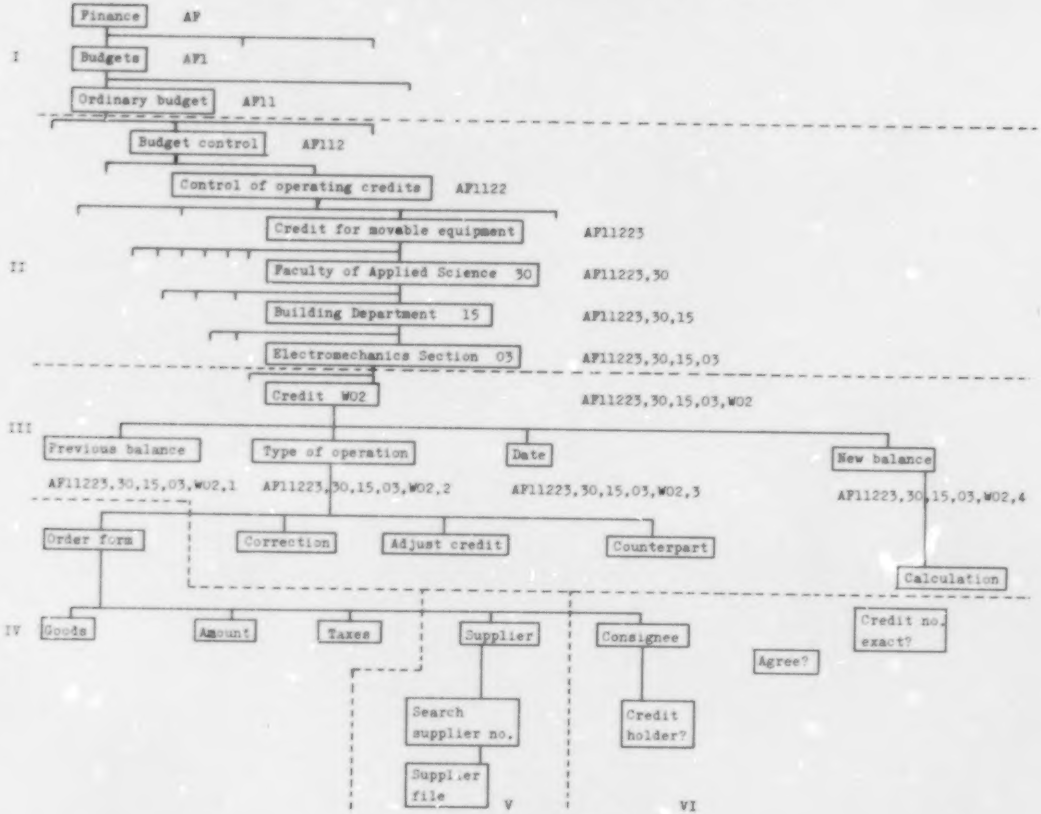
For example, for administration of wages and salaries, we can distinguish:

- updating of the data base files
- monthly calculation of wages and salaries
- monthly correction after calculation
- monthly closing and accounting
- quarterly corrections
- quarterly adjustments (Social Security)
- legal holiday allowances
- annual corrections
- annual adjustments (taxes...)
- drawing up the personnel budget
- control of the personnel budget.

We compare this subdivision with the logical subdivisions of the activities diagram. We can adopt these information groups in the same way in our activities diagram.

Figure 8a

DETAILED ACTIVITIES PROGRAM DIVIDED INTO LOGICAL SETS


Finance AF
I
Budgets AF1
Ordinary budget AF11
Budget control AF112
Control of operating credits AF1122
Credit for moveable equipment AF11223
II
Faculty of Applied Science 30 AF11223,30
Building Department 15 AF11223,30,15
Electromechanics Section 03 AF11223,30,15,03
Credit W02 AF11223,30,15,03,W02
III
Previous balance AF11223,30,15,03,W02,1
Type of operation AF11223,30,15,03,W02,2
Date AF11223,30,15,03,W02,3
New balance AF11223,30,15,03,W02,4
Order form
Correction
Adjust credit
Counterpart
Calculation
IV
Goods
Amount
Taxes
Supplier
Consignee
Agree?
Credit no. exact?
Search supplier no.
Supplier file
Credit holder?
V
VI

Finally, if we take the entry input points of each of these groups and introduce them into the activities diagram, we can conclude that we have a logical data base.

However, we should not forget that in many cases adjustment of the information at operational level must also be reproduced in summary form at management level.

3. Establishment of the Physical Data Base

We now have all the necessary elements for the last step, which is the definite establishing of the physical data base. We know the hierarchy of information units; they are subdivided into usable groups. We have adjusted these groups to the procedures to be followed. We know the input entry points. We know how frequently the information units are used by the different groups. Being in possession of all these data, we choose the most effective solution, which is then recorded in the Data Dictionary.

In conclusion, we shall now compare the established data base with the real programs. This "efficiency test" which must go on is one of the tasks of the data base administrator. We have tested the previous technique in our information system and we shall analyse in the future several global systems.

It is clear that this system is not free of criticism. We are therefore asking the "Data Base and Management" working group to carry out a feasibility study on this system as it applies to non-university administrators.

4. A DATA DICTIONARY/DIRECTORY

We will describe the automated Data Dictionary System that has been developed at the K.U. Leuven. This system uses in-house standards and has been a tool in the method of analysis which we have been developing at the University since 1972.

4.1 Problem Definition

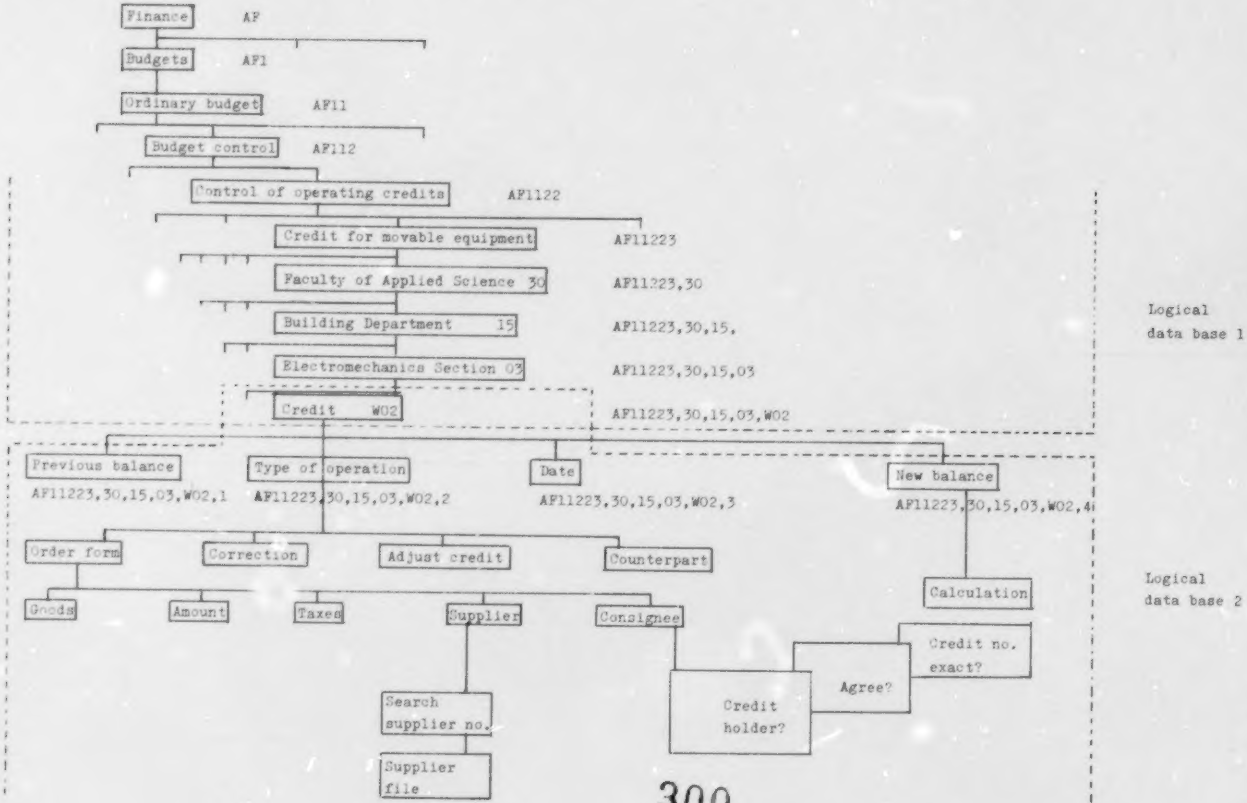
Our study started in 1972. At that time we already felt that there were some shortcomings in the traditional analysis when used in a data base environment for the implementation of an Information Management System.

Until now, in what we call the classical systems, all information concerning automatic data processing is recorded in a documentation file. The documentation tasks have been standardised as much as possible and various attempts to automatise have been made.

In most cases, the part on "Description of Data" in the project documentation is made available in duplicated form to the persons concerned. It is likely that the greatest part of the program file

Figures 8b and 8c

DETAILED ACTIVITIES DIAGRAM


Finance
AF
Budgets
AF1
Ordinary budget
AF11
Budget control
AF112
Control of operating credits
AF1122
Credit for movable equipment
AF11223
Faculty of Applied Science 30
AF11223,30
Building Department 15
AF11223,30,15,
Electromechanics Section 03
AF11223,30,15,03
Credit W02
AF11223,30,15,03,W02
Logical data base 1
Previous balance
AF11223,30,15,03,W02,1
Type of operation
AF11223,30,15,03,W02,2
Date
AF11223,30,15,03,W02,3
New balance
AF11223,30,15,03,W02,4
Order form
Correction
Adjust credit
Counterpart
Calculation
Logical data base 2
Goods
Amount
Taxes
Supplier
Consignee
Credit no. exact?
Agree?
Credit holder?
Search supplier no.
Supplier file

Figures 8d, 8e, 8f, 8g, 8h and 8i

EXAMPLES OF PHYSICAL DATA BASES EXTRACTED
FROM THE ACTIVITIES DIAGRAM (Figure 8a)

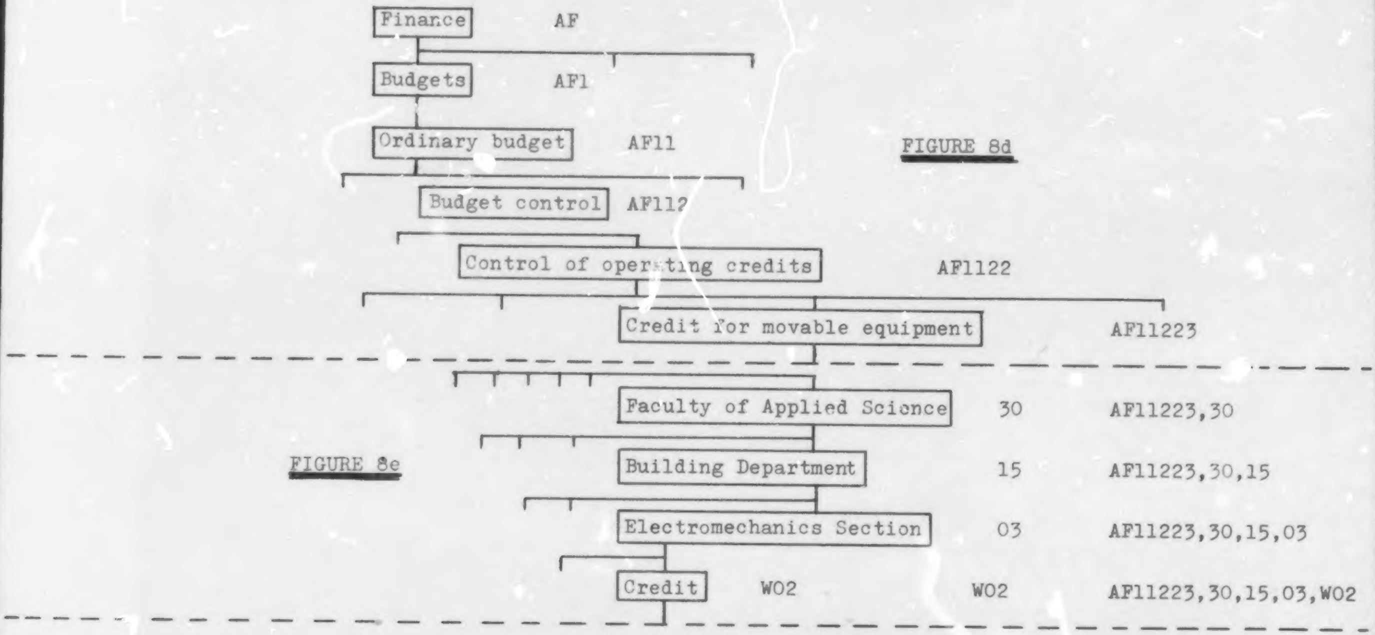
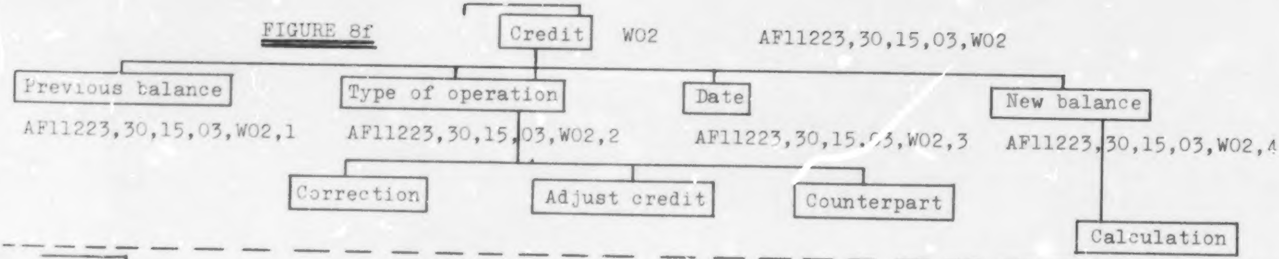

Finance AF
Budgets AF1
Ordinary budget AF11
Budget control AF112
Control of operating credits AF1122
Credit for movable equipment AF11223
Faculty of Applied Science 30 AF11223,30
Building Department 15 AF11223,30,15
Electromechanics Section 03 AF11223,30,15,03
Credit W02 W02 AF11223,30,15,03,W02

FIGURE 8d

FIGURE 8e

FIGURE 8f


Credit
W02
AF11223,30,15,03,W02
Previous balance
AF11223,30,15,03,W02,1
Type of operation
AF11223,30,15,03,W02,2
Date
AF11223,30,15,03,W02,3
New balance
AF11223,30,15,03,W02,4
Correction
Adjust credit
Counterpart
Calculation

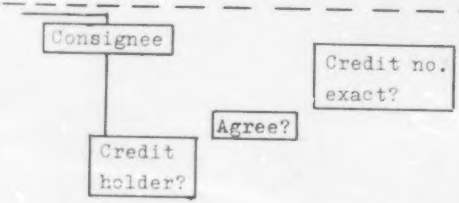
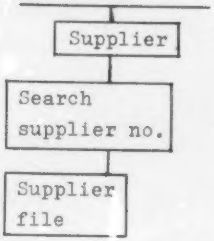

Consignee
Credit no. exact?
Agree?
Credit holder?

FIGURE 8g

FIGURE 8i


Supplier
Search supplier no.
Supplier file

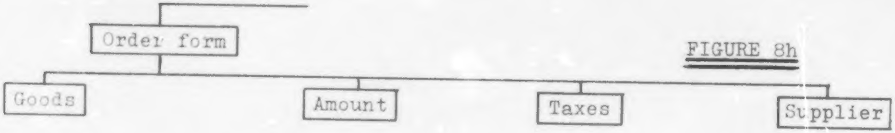

Order form
Goods
Amount
Taxes
Supplier

FIGURE 8h

will be duplicated, with the exception of the programs themselves. As a result, however, it is very difficult to modify all the copies in circulation and after a while there may be great differences among them. Moreover, such a method does not provide replies to such questions as "where else is the data element used?" or "who is allowed to revise a data element?".

When data bases are used, the description becomes more voluminous and even more difficult to keep up to date. As a matter of fact, we then have to take into consideration the links between data elements; the description of the structure itself becomes a necessity.

These considerations lead us to conclude that we need a more efficient and, most likely, automated means to fulfil the above-mentioned tasks and needs. Finally, a data base system requires standardisation and a more elaborate documentation.

4.2 Principles and Design of the DD/D at the K.U. Leuven

The DD we have is a common Data Dictionary for all projects. The live dictionary as well as the development dictionaries are in a central and common data base.

In a first phase (up to now) we have been developing the DD like an informal dictionary. In a second phase (1976), we shall develop it into an operational (executable) dictionary.

It is a single-entry system. The system elements are defined once and referenced where necessary. The upward references are automatically made. The indexing is as follows: A KWIC (Key Word In Context) combined with item reference indexing.

Standards

First of all we developed a number of standards which are to be used in analysis and programming. These are the "System's Names" for the different data elements used by the DD/D. Among the data elements used are:

- systems
- runs
- program names
- files
- records
- data base names
- segments
- PCB's (Program Control Block)
- fields

The standard conventions are described in Figure 9.

Figure 9

K.U.L.STANDARD CONVENTIONS	1 2 3 4 5 6 7 8	D DD/IDENT	SS PROJECT IDENT	T TYPE IDENT	N SERIAL NO.	A	WHERE USED?
RUNS	D S S T N N N V ← DD/USAGE → ← PROG/USAGE →	- E	example BK Bookkeeping	T - TEST RUN P - if 1 prog. in that RUN R - more than 1 prog.	A unique identifier in a project		In JCL. In Data dictionary
PROGRAM-NAME	D S S N N N ← PROG/USAGE → ← DD/USAGE →	-P	BK		A unique identifier in a project		- PROGRAM-ID of COBOL - LINK.LIBRARY - as PSB-name - as stepname in JCL - In DATA dictionary
FILES	D S S T N N N V ← PROG/USAGE → ← DD/USAGE →	-B	BK	F - Fixed W - Workfile	A unique identifier in a project		- In SELECT clausule as DD name - In JCL as DD-name and DS-name - As label on Disk and Tape - in Data dictionary
RECORDS	D S S T N N N ← DD/USAGE → ← PROG/USAGE →	-R	BK	C - CARD R - RECORD/ TABLES P - PRINTER	A unique identifier per type/project		- In the COPY LIBRARY - As PREFIX for the DATA ELEMENTS in the programs. - In the Data dictionary
DATA ELEMENTS	D T A A A A A A ← DD/USAGE → ← PROG/USAGE →	-V		See Appendix(a)		A unique Keyword description (&N)	- As Field name of the COPY'S - As Field name in the programs - In the Data dictionary
D B - NAME	D S S T N N ← DD/USAGE → ← IMS/USAGE →	-D	BK	F - Physical DB L - Logical DB	A unique identifier in a project		- In the DBD LIBRARY (IMS) - In the PCB'S (IMS) - In the logical and physical DBD'S (IMS) - In the Data dictionary
SEGMENT-NAME	D T N N N N ← IMS PROG/USAGE → ← DD/USAGE →	-S		F - Physical P - Path-call L - Logical	A unique identifier per segment type		- In the physical and logical DBD'S (IMS) - In the PCB'S (IMS) - In the COPY LIBRARY (COPY of each segment) - As prefix for the data elements - In the DD
PCB-NAME	D S S N N T M M ← DD/USAGE → ← IMS/USAGE →	-C	BK	T - Terminal PCB D - DB-PCB	Same serial No. as program-name		- In PSB LIBRARY (IMS) - In the Data dictionary

NB: V = free position for additional information in the case for RUNS & FILES
example: File which is I or 0...

Appendix

THE STUDENT INFORMATION SYSTEM AT THE K.U. LEUVEN

1. Aims

When developing the MIS system, the steering group decided to integrate the Student Information System into the global management system of the K.U. Leuven. Integration offers the following possibilities:

- access to all data in the Student Information System;
- access to all data of this system after it has been linked to other parts of the greater system.

This information is therefore accessible to all departments at the K.U. Leuven insofar as access is required in the performance of their functions.

The global Management Information System of the University, of which the Student Information System forms part, can be used thanks to a new software permitting the use of a data base.

This software also allows us to secure a real integration of the various subsystems (of which the Student Information System forms part). Such integration greatly increases the importance and the value of data processing.

2. Organisation and Development of the Student Information System

The steering group set up a student project group consisting of members of the university departments dealing with student information in the performance of their duties. Such project group members included representatives of:

- academic and scientific personnel
- student welfare services
- planning office
- student administration office
- faculties
- monitors and assistants.

This student project group was responsible for the progress of activities. In view of the complexity and the problems related to the development of a Student Information System, two sub-groups were set up. One of these groups was in charge of collecting information required for such a system. The second group was responsible for studying and describing possible procedures for the new enrolment system and its requirements (types of information, timing..).

The activities of both sub-groups can be summarised as follows:

a) Collection of data elements for the Student Information System:
  - definition of the type of information
  - determination of the timing of collection
  - determination of the place of collection
  - determination of the way of collection
  - consideration of organisational aspects related to collection:
    - technical organisation
    - control of collected data
    - personnel organisation.

b) Processing of Student Information:
  - development of codification
  - determination of control
  - planning the processing
  - frequency of processing.

c) Dissemination of student information:
  - who can, may or has to get this information?
  - when must the information be available?
  - in what form must the information be provided?

At the end of the study, it was decided to give a priority to the automation of the student enrolment system (a subsystem of the Student Information System). By now, the automation of this enrolment system is almost completed.

3. Content of the Student Information System

The Student Information System is composed of:

a) A current and a historical data base

The current data base includes information relating to students currently enrolled at the University, regardless of the fact that they were enrolled in previous years.

This data base is composed of 9 segments:

1. Fixed identification elements. This segment consists of the following records:
   - date and place of birth
   - sex
   - surname and first name
   - registration number (which the student keeps throughout his university studies)
2. Variable elements
   - parents' address
   - address of residence

- telephone number of both addresses
    - nationality
    - civil status
3. Elements related to enrolment at the K.U. Leuven
    - year of enrolment
    - enrolment fee
    - examination results
    - additional information regarding the year of study
4. Elements related to secondary education
    - field of studies
    - results obtained
    - addresses of educational institutions
5. Elements related to post-secondary education
    - field of studies
    - results obtained
    - addresses of educational institutions
6. Elements related to the social environment of the student
    - occupation of parents and grand-parents
    - family situation
7. Elements related to married students
    - particulars of spouse
    - accommodation
8. Elements related to means of transport
    - means of transport to and from Leuven
    - means of transport in Leuven
9. Elements related to financing of studies
    - scholarships
    - loans

The historical data base contains information regarding students who have studied at the University in the past but who are not currently enrolled. The structure and content of the historical data base are similar to those of the current data base.

b) Municipal Data Base

This data base contains information regarding the Belgian municipalities. It is also used by other subsystems of the MIS.

c) Year of Study Data Base

It contains information per year of study and is also used by other subsystems of the MIS.

d) Educational Institutions Data Base

It contains all universities, post-secondary and secondary educational institutions in Belgium.

e) A Number of Classical Files

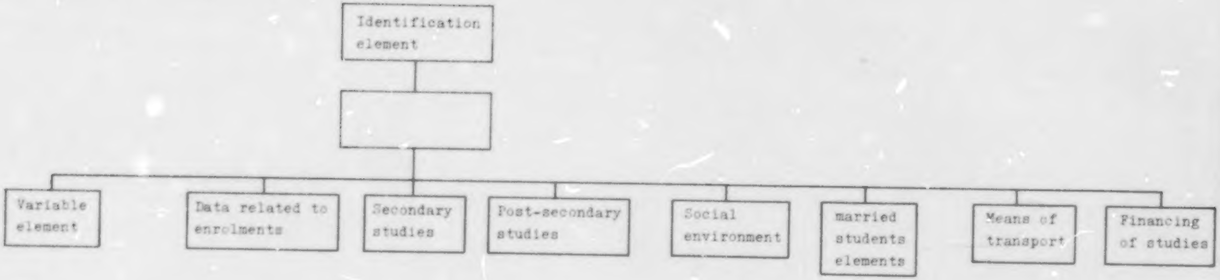
- secondary education diploma
- post-secondary education diploma
- nationality
- districts and provinces
- streets of Leuven and surroundings
- faculties and departments
- occupation
- level of study

4. Current Output of the Student Information System

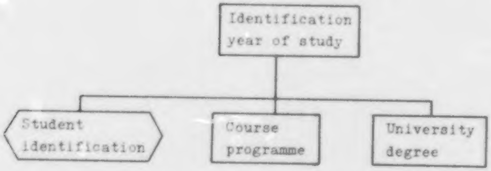
1) A number of statistics with unlimited possibilities.
   For example:
   - new students by district and province
   - list of all students by nationality, sex and field of study
   - examination results
2) Listings with information on enrolled students
3) Attestations
4) Listings for University top management.

Figure 10

STUDENT DATA BASE


Identification element
Variable element
Data related to enrolments
Secondary studies
Post-secondary studies
Social environment
married students elements
Means of transport
Financing of studies

YEAR OF STUDY DATA BASE


Identification year of study
Student identification
Course programme
University degree

SELECTED ANNOTATED BIBLIOGRAPHIES

SELECTED ANNOTATED BIBLIOGRAPHIES TO CHAPTER 2

A. ON BASIC CONCEPTS OF INFORMATION SYSTEMS

Aaron, J.D. (1969) "Information Systems in Perspective", Computing Surveys, I, No. 4, pp. 213-36. An excellent survey of the field with special emphasis on MIS (Management Information Systems). It covers communications and graphics. It is addressed to the computer scientist but is not too technical.

Bittel, L.R. (1969) Management by Exception, New York: McGraw-Hill Book Company, Inc., 320 p. A business-oriented book concerned primarily with the concept of management by exception. In the final chapter, the author relates the exception concept to operations research and information systems.

Churchman, C.W. (1968) The Systems Approach, New York: Dell Publishing Co. Inc., 243 p. This extremely readable text contains excellent discussions on "systems" thinking, efficiency, and some basic considerations for systems. Included is an example of how management scientists incorporate models and simulation techniques to aid in decision-making. Further, the discussion of management information systems gives broad guidelines for feasibility, reliability of data, security, and validity of user's requirements. An annotated bibliography is included.

Crowley, W.J. (1966) "Can We Integrate Systems Without Integrating Management", Journal of Data Management, IV, August, pp. 14-18, 23-24. Mr. Crowley believes that it doesn't make much difference how we answer the question "Can we integrate systems without integrating management?" We are going to progress toward integrated systems, i.e., systems which will: (1) supply historical data and analysis of that data; (2) supply "on-line" data; that is, factual material picked right out of the system as fast as it is generated, and (3) supply data in "real time" fast enough so that management can exercise necessary management control instantly. This well-written and informative article also covers the place of the EDP specialist in top management, three kinds of management decisions (operational, tactical and strategical), and what the author terms as

unanswerable questions beyond the total integrated system. Though this article is written for the businessman it still has validity for the educational administrator.

Davis, G.B. (1974) Management Information Systems: Conceptual Foundations, Structure, and Development, New York: McGraw-Hill Book Company. An extensive discussion on concepts. A good introductory text on information systems.

Harvey, A. (1964) "Systems Can Too Be Practical", Business Horizons, VII, No. 2, Summer, pp. 59-69. The author presents and explains the following practical shortcuts that make the application profitable in many common business situations: (1) a realistic systems approach does not have to take into account every contingency, (2) both short and long-term variables can be identified, (3) the system can live with "real-life" limitations, and (4) practically applied, the systems approach can be implemented step-by-step. Management attitude is, in the opinion of the author, the largest factor in system failure or success.

Head, R.B. (1967) "Management Information Systems: A Critical Appraisal", Datamation, XIII, No. 5, pp. 22-27. Mr. Head is the founder of Software Resources Corporation and is a frequent contributor to the literature concerning the data-processing field. He provides an imaginative conceptual framework of the hierarchy of the data base for management information. The author emphasizes that system designers must come to grips with problem areas concerning the levels of the data base, specialised data bases, and management's "information threshold". An understanding of these problems is needed not only by system designers, but also by management. Industry trends are identified and briefly discussed under the headings of: application reworking, formalised systems planning, management science emphasis, regional information processing, and information management systems.

Hockman, J. (1963) "Specifications for our Integrated Management Information Systems", Systems and Procedures Journal, XVIII, No. 5, pp. 40-41. Systems designers desiring general statements concerning the objective, scope, and content of an integrated management information system will find this article rewarding reading. The main points are that the system should provide only relevant information in the required amount, and the system should be capable of evaluating and redefining objectives.

Johnson, R.A., F.E. Kast and J.E. Rosenzweig (1967) The Theory and Management of Systems (2nd ed.), New York: McGraw-Hill Book Company, Chap. IV, pp. 71-91. The authors have blended their academic experience gained as members of a respected graduate school of business administration with actual operational situations to present a very informative and very readable treatment of the systems concept. Chapter 4 provides readers with a clear and non-technical explanation of the control and systems concept. Control functions include the measurement of output, the comparison of output with some predetermined standard, and the adjustment of inputs to restore the system to its planned norm. Included is a section on the applications of control systems in an educational system.

Keller, J. (1969) "Higher Education Objectives: Measures of Performance and Effectiveness", in eds. **J. Minter and B. Lawrence**, Management Information Systems: Their Development and Use in the Administration of Higher Education (2nd Ed.). Boulder, Colo.: Western Interstate Commission for Higher Education, pp. 79-84. The author exhibits four parameters which can be used to study the causes of poor performance at the university level. Keller states that a better understanding for university outputs, the relationships between them, and the costs involved could be achieved through a formal analysis of the chosen parameters.

Rourke, F.E. and G.E. Brooks, (1966) The Managerial Revolution in Higher Education, Baltimore, Md.: The Johns Hopkins Press, Chap. III, pp. 44-67. This is a survey of 120 universities' use of institutional research. The questions surveyed include: who is most influenced by institutional research; who controls the projects; what projects are undertaken; and organisational problems. The authors predict that the primary mission of institutional research will become the study of academic programs and policies.

In addition to the above, there are many texts and books that discuss basic concepts in information systems. An annotated bibliography of these books follow. They are organised in two sections:

(1) Computer Science and business oriented
(2) Educational Administration oriented

B. ON COMPUTERS AND INFORMATION SYSTEMS

Blumenthal, S.C. (1969) Management Information Systems: A Framework for Planning and Development. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 219 p. Presents a formal framework for information systems and is recommended to the serious reader.

Davidson, C.H. and E.C. Koenig, (1967) Computers: Introduction to Computers and Applied Computing Concepts. New York: John Wiley & Sons, Inc., 596 p. An introduction to computers and computer programming (FORTRAN). Discusses many applications also.

Desmonde, W.H. (1964) Computers and Their Uses. Englewood Cliffs, N.J.: Prentice-Hall, Inc. 296 p. This is a good overview of computers and their implications. It includes a discussion of punched-card machines, machine logic, concepts of programming, applications, and social considerations of computers.

Heany, D.F. (1968) Development of Information Systems: What Management Needs to Know. New York: The Ronald Press Co., 421 p. An excellent text on information systems and is written in non-technical terms.

Kanter, J. (1967) The Computer and the Executive. Englewood Cliffs, N.J.: Prentice-Hall Inc., 134 p. It is successful in its claim of exploring the computer world from the management viewpoint "in terms familiar to the executive or the aspiring executive".

McDonough, A.M. and L.J. Garrett, (1965) Management Systems Working Concepts and Practices. Homewood, Ill.: Richard D. Irwin, Inc., 325 p. A good text for an information system in the business and industrial environment.

Meadow, C.T. (1967) The Analysis of Information Systems: A Programmers Introduction to Information Retrieval. New York: John Wiley & Sons, Inc., 301 p. This is a technical book but is excellent on the design of an information retrieval system and an introduction to data structures.

Prince, T.R. (1966) Information Systems for Management Planning and Control. Homewood, Ill.: Richard D. Irwin, Inc. The text is applications-oriented and has many case studies from industry.

Sisson, R.L. and R.G. Canning (1967) A Manager's Guide to Computer Processing. New York: John Wiley & Sons, Inc., 124 p. These authors have written a series of books for managers. This one is "management-oriented" explanation of information systems and computers and their role in managing a company.

C. ON SYSTEMS APPROACHES TO EDUCATIONAL MANAGEMENT

Banghart, Frank W. (1969) Educational Systems Analysis. London: Macmillan and Co. Ltd., 330 p. The best book in the field because it is the only one. It has a collection of Operational Research techniques that can be found in many other textbooks. It has two chapters on information systems design but these are brief and mostly outdated. But the book is a good attempt at integrating systems techniques to education and has an interesting chapter on Scientific Administration. The 1969 edition has numerous printing errors including the omission of code numbers which makes ten pages on the Planning and Control chapter totally worthless.

Bushnell, Don D. and Dwight W. Allen, eds. (1967) The Computer in American Education. New York: John Wiley & Sons, Inc., 300 p. This book is a set of readings by educators and computer scientists. The topics discussed include: individualised instruction and social goals, computers in instruction and research, teaching the computer sciences, information processing for education systems. It discusses the potentialities and limitations of computer technology in education.

Gerard, Ralph W., ed. (1967) Computers and Education. New York: McGraw-Hill Book Company, Inc., 307 p. This book is a collection of papers and discussions of a workshop that took place in the fall of 1963. As such, it is outdated. But it does have good discussions on library automation, CAI (Computer Aided Information), administrative applications, as well as regional and national networks.

Goodlad, John I., J. F. O'Toole, Jnr. and Louise L. Tyler (1966). Computers and Information Systems in Education. New York: Harcourt, Brace & World, Inc., 152 p. The book gives an overview of Education and Computer Technology; EDP in Education, present developments and technical aspects; the state of the art; applications of computers and information systems in education; and discusses problems, issues, and recommendations pertaining to electronic data systems.

Grossman, Alvin, and Robert L. Howe, (1965) Data Processing for Educators. Chicago: Educational Methods, Inc., 362 p. Covers history of automated information systems; developing an educational data processing system; equipment, personnel, and design requirements; typical educational applications; impact of EDP on schools; and the potential for education. It has many samples of output and is largely school oriented.

Gruman, Allen J. (1965) Workbook: Data Processing for Educators. Chicago: Educational Methods, Inc., 138 p. This workbook is designed to be used with Grossman and Howe, Data Processing for Educators, provides exercises covering costs of equipment; pay scales; machines and their functions; systems study; systems design; punch card design; scanner forms design; paper forms design; test scoring; attendance records and accounting; student scheduling; and cumulative records.

Haga, Enoch, Ed. (1967) Automated Educational Systems Elmhurst, Ill.: The Business Press, 343 p. This book is a series of readings on school data processing. It is in two parts. The first part discusses "concepts and patterns" with a heavy emphasis on regional and national educational information systems. The second part discusses applications and techniques including the CARDPAC and Plato Systems.

Hartley, Harry J. (1968) Educational Planning-Programming-Budgeting: A Systems Approach. Englewood Cliffs, N.J.: Prentice-Hall Inc., 290 p. Topics discussed are educational planning and economic rationality; overview of systems analysis in education; PPBS; need for budgetary reform; program structures and economic issues in schools; administrative issues and implications of PPBS.

Hussain, K.M. (1973) Development of Information Systems for Education, Englewood Cliffs: Prentice-Hall, Inc. Hussain.... intends the book to be a major vehicle by which top administrators can develop their knowledge and interest in IS and thus contribute their support to systems development.... Hussain's book is especially well written and consequently easily read and digested.... Throughout the book charts, graphs, illustrations and example facilitate the understanding of information systems. Robert J. Barak in College and University Business.

Johnson, C.B. and W.G. Katzenmeyer, eds. (1969) Management Information Systems in Higher Education: The State of the Art. Durham, N.C.: Duke University Press, 191 p. A set of readings including discussions of a "Theory of University Management":

national (HEGIS) and regional (WICHE) systems; and modelling in management of information systems.

Kalmann, Richard A. and Robert W. Marker, (1967) Educational Data Processing: New Dimensions and Prospects. Boston: Houghton Mifflin Co., 326 p. Perhaps the best overall book in educational data processing. Contains fifty different articles covering the field of computers in education. Discusses computer concepts specifications; personnel required; who should control; centralisation vs. decentralisation; applications; starting an EDP program; determining need for data processing; simulation; and data banks.

Kershaw, J.A. and R.N. McKean (1959) Systems Analysis and Education. Santa Monica, Cal.: RAND Corporation, 64 p. This old book in its field is a research memorandum-working paper, and contains formal quantitative comparisons of specific education systems with variants of them in which changes and innovations are incorporated. This memorandum is concerned with elementary and secondary schools only.

Loughary, John W. (1966) Man-Machine Systems in Education. New York: Harper and Row, 242 p. This book covers such topics as: computers and instruction; instructional MIS (Management Information Systems); computers in school management; applications to state departments of education; district applications; educational policy development; computers for pupil personnel services; and implementing man-machine systems.

Pfeiffer, John, (1968) New Look at Education. New York: The Odyssey Press, 162 p. A paperback written in non-technical terms. It emphasizes the roles of models in educational decision-making and gives numerous samples from schools and colleges.

Systems Analysis for Educational Planning. (1969) Paris, France: Organisation for Economic Co-operation and Development, 219 p. An excellent annotated bibliography of the literature from the United States and abroad. Has 306 entries subdivided into nine sections including selected techniques applied to education, data processing and information systems for education, and studies in selected topics of systems analysis.

UNESCO. (1964) IIEP, Educational Planning, A Bibliography. Paris, France: UNESCO 131 p. Somewhat outdated yet a good collection of 540 entries. The entries are annotated and cover the literature from around the world. The material reviewed includes cases and educational planning, as well as organisation and administration.

Whitlock, James W. (1964) Automatic Data Processing in Education. New York: The Macmillan Co., 144 p. Application to local school districts, state departments of education and institutions of higher learning; student personnel accounting; school business and personnel functions; and installation planning. This book is largely EAM equipment oriented and is, therefore, largely outdated.

ANNOTATED BIBLIOGRAPHY TO CHAPTER 3

Blumstead, A.R., (1967) "The Designs Approach to an Educational Data System", AEDS Journal, No.2, pp.57-66. A good overview of the development process applied specifically to education.

Kirk, F.G., (1973) Total System Development for Information Systems, (New York: John Wiley and Sons), Ch. 4, pp.31-40. Has an excellent diagram showing the interrelationship of the different phases of development. Each of these phases are discussed in a full chapter subsequently.

McDonough, A.M. and L.J. Garrett, (1965) Management Systems: Working Concepts and Practices, (Homewood, Illinois: Richard D. Irwin, Inc.), Ch.XV, pp.199-209. This is an excellent review and summary of management systems. Systems may be thought of as ladders that ease the climb toward goal accomplishments and that effective management information systems can alleviate the impact of organisational and managerial changes. A summary diagram for management systems and the accompanying explanation provide a review of the authors' concept of management information systems.

Orlicky, J., (1969) The Successful Computer System, (New York: McGraw-Hill Book Company), Ch. VII, pp.135-150. A good summary and review of the information system development process. The "phases" discussed are: programming, supporting procedures, education, conversion planning and development of auxiliary procedures, file cleanup, program testing, debugging, and conversion.

ANNOTATED BIBLIOGRAPHY TO CHAPTER 4
ON DESIGN OF DATA BASES

This annotated bibliography is an attempt to guide a reader through the literature on Data Base Design so as to enable him to select according to his interests, needs and background.

Some students helped in the literature search and in preparing some of the annotations. These are D.L. De Reis, W.S. Davidson and D. Molleur. Their contributions are herewith acknowledged.

Bachman, C.W. (1969), "Data Base Management: The Keystone of Applied System Architecture". In Gruenberger, F., *Critical Factors in Data Management*. Englewood Cliffs: Prentice-Hall, pp. 33-40. The article is an overview of the subject; the author gives definitions and explanations of job scheduling, recovery and restart. The article is completely user-oriented.

Bayless, D. and D. Shuman, (1973), "Protect Your Data Base", *Computer Decisions*, Vol. 5, No. 10. Increasing demands for reliable back-up systems which will safeguard data integrity are causing many administrators to take a more serious attitude in their selection of such systems. Three back-up and recovery plans are discussed (and compared) in this article. Rerun (re-introducing input), update record method (passing the back-up file against the master file dump), and edited input (saving updated fields) are the three examples used. The methods are evaluated in terms of programming effort, file dump frequency, machine requirements and the required volume of back-up data.

Byrnes, C.J. and D.B. Steig, (1966), "File Management Systems: A Current Summary", *Datamation*, Vol. 15, No. 1, pp. 138-142. Addressed to management, it covers definitions, uses ("why have one?"), and descriptions of existing file management systems. Article has a chart which allows rapid comparison of existing commercial packages. It is a "consumers guide" of commercial file management systems.

Brewer, R.J. (1973), "Waiting for the Data Base". In Bassler, R.A. and Logan, J.J., *The Technology of Data Base Management Systems*, Arlington: College Readings, Inc., pp. 39-40. The article explains the MIS of the Missouri Division of Mental Health (DMH).

The author notes central staff co-ordination, use of programming resources and retrieval systems used in the system.

Cahill, J.J. (1970), "A Dictionary/Directory Method for Building a Common MIS Data Base", Journal of Systems Management, Vol. 21, No. 11, pp. 1-14. Defines the characteristics of a common data base. Encompasses the definition of guidelines for establishing information for a common data base, data specifications and specification statements for data, use of the dictionary/directory, and also through the maintenance and updating of the dictionary/directory. (The article is derived primarily from experience and methods developed by IBM.)

Canning, R.G. (1974a), "The Current Status of Data Management", EDP Analyser, Vol. 12, No. 2(a), pp. 1-14. This article gives an overview of the current data management ideologies. The article briefly discusses specific new data management systems including System 2000, Arcata Data Management, ADABAS, and others. The inverted file technique is discussed in detail. This article is part of a series on Data Base Management.

Canning, R.G. (1974b), "The Data Dictionary/Directory Function", EDP Analyser, Vol. 12, No. 11. An excellent survey of the current state of the art. Also a very good discussion of many of the existing DD/D packages: LEXICON, Three D System, UCC TEN, DATA CATALOGUE, IBM IMS Dictionary System, PRIDE-LOGIC, and MPS DATAMANAGER.

CODASYL Systems Committee (1971), "Introduction to 'Feature Analysis of Generalised Data Base Management Systems'", Communications of the ACM, Vol. 14, No. 5, pp. 308-318. This introduction is an excellent preface to the feature analysis; however, in addition to being an introduction it also functions as a very good glossary of terms and, in general, a good work in its own right. It includes not only detailed explanations of features being analysed, but also provides insight to recommendations for further needed study. The task group itself is involved in building more advanced techniques for processing data and seems to find COBOL as its basic foundation for implementing many of its suggested features. The feature analysis compares nine different generalised data base management systems and not only compares the general features of each system but also a good theoretical background and analysis for its discussion. The report identifies four levels of users: the administrator, programmer, non-programmer and parametric user.

Collmeyer, A.J. (1971), "Data Base Management in a Multi-Access Environment", *Computer*, Vol. 4, No. 6, pp. 36-46. The author defines a Data Base Management System as a network of logical subsystems which perform a specific function in the network. These relationships between functional subsystems are carefully detailed and explained.

Coulouris, G.F., J.M. Evans and R.W. Mitchell, (1972), "Towards Content-Addressing in Data Bases", *The Computer Journal*, Vol. 15, No. 2, pp. 95-98. An overview of existing Data Base management systems. Proposal of a hardware-aided content address system. Rather technical.

Cuozzo, D.E. and J.F. Kuntz, (1973), "Building a Base for Data Base: A Management Perspective", *Datamation*, Vol. 19, No. 10, pp. 71-75. The purpose of this article is to help potential users make a Data Base Management System go or no-go decision. The authors present key features, benefits, and disadvantages of implementing a DBMS, and advise managers to thoroughly plan and prepare for future implementation of the system. It examines some of the serious considerations a MIS/EDP manager must undertake concerning the one-time and ongoing incremental cost of a Data Base Management System (DBMS). It also provides significant insight into the reduction of cost by initiating planning and preparation programs for both future data base facilities and possible short-term alternatives.

Curtice, R.M. (1974), "Some Tools for Data Base Development", *Datamation*, Vol. 20, No. 7, pp. 102-106. This article is an overview including a discussion of the data Dictionary, table driven editor, and TEST data base generator. It describes basic requirements, advantages and disadvantages of these tools. The author explains the importance of these tools during development, testing and after implementation of the system. An excellent introduction to data base design.

Dodd, G.G. (1969), "Elements of Data Management Systems", *Computer Surveys*, Vol. 1, No. 2, pp. 117-133. This paper is concerned with data structures such as sequential, random and list organisations and a look at such complex structures as multi-list files, indexed sequential files, tree structures, rings and hierarchical structures. An excellent introductory non-technical survey of data structures.

Dyba, J.E. (1973), "Benefits of Data Base and Doing one Better", *Data Management*, Vol. 11, No. 3, pp. 31-32. The author stresses two main benefits of data bases - capability and flexibility.

Also discussed are reduction of redundancy, independence, integrity, security, changeability and ease of use. He recommends the use of interfaces to increase flexibility.

Goddard, S., J. S. Martin and L.C. Romney, (1973), Data Element Dictionary, Second Edition, Technical Report No. 51, Boulder, Colorado: NCHEMS at WICHE. This is a set of five DED's. One each for the files of Courses, Facilities, Finance, Staff and Students. The data elements listed therein are required for implementing models developed by NCHEMS, the National Center for Higher Educational Management Systems in the United States. They do not include all data elements required by institutions for their operational data base. An appendix includes the HEGIS taxonomy: the Categories for Current Funds Revenue Functional Classifications and Standard Room Uses; formats for calendar dates: abbreviations for states in the United States and countries of the world; and visa types. These DED are to be revised periodically.

Goldstein, R.C. (1974). "Tutorial on Computer Data Bases", in Nolan, R.I., Managing the Data Resource Function, New York: West Publishing Company, pp. 349-389. This article examines five main areas of concern pertaining to this subject. The role of the data base in the organisation, technical aspects of organising information, interfacing between the data base and users, and privacy and security are discussed in some detail. Also discussed are considerations of file design, file structure, data base organisation and addressing techniques, and access mechanisms. It is also good on physical data structures. An excellent tutorial.

Gruenberger, F. (Editor)(1969), Critical Factors in Data Management, New Jersey: Prentice-Hall, Inc., 146 pp. Selected works from a symposium on Critical Factors in Data Management/1968. Contains a cross-section of papers directed toward problems in management information systems, data base management, and time-sharing systems. Written in relatively technical terms; it provides, however, some interesting views of some of these factors and some possible solutions or alternatives.

Gulko. W.W. (1972). Program Classification Structure, First Edition, Technical Report No. 27, Boulder, Colorado: NCHEMS at WICHE, 136 pp. This is an authoritative source on an important set of data elements used in higher education: The Program Classification Structure (PCS). In it is contained not only the structure and its many subclassifications used in academic and non-academic sectors, but also a good discussion of the rationale and use of the PCS.

Harrell, C., Jr. (1973), "Maintaining a Healthy Data Base", in **R.A. Bassler and J.J. Logan**, The Technology of Data Base Management Systems, Arlington: College Readings, Inc., pp. 103-104. The author states there is, currently, little concern about the reliability of data after creation of the data base. He suggests that constant testing and examination of data is essential to the well-being of the data base. Regular managèment auditing reviews should be implemented.

Harvison, C.W. and K.J. Radford, (1972), "Creating a Common Data Base", Journal of Systems Management, Vol. 23, No. 6, pp. 8-12. The process of consolidating existing data files within an organisation and the subsequent creation of a supporting file directory system are covered by this article. The article extols the merits of a gradual, systematic conversion accompanied by a good classification of data elements. This classification process will follow the functional structure of the organisation, simply because it becomes relatively independent of organisational changes (the functional structure should change only if the objectives of the enterprise itself changes). The authors stress management co-operation. They advocate the "evolutionary approach" of conversion to a common data base in order to avoid disrupting the organisation.

Hussain, K.M. (1973), Development of Information Systems for Education, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 419 pp. The only book on the subject that is specifically addressed to the educational administrator and manager. It does not discuss dictionaries or directories, but does discuss data hierarchy and the design of a data base as part of an information system. Thé author makes extensive use of PERT diagrams to discuss each stage and activity required in the development of an information system. Has numerous illustrations from processing in colleges and universities.

IBM (1970), Introduction to Data Management, Student text SC 20-8092-0, New York: IBM Corporation, 53 pp. The text provides a basic explanation of terms and subjects relating to data management. Data storage devices, data organisation and functions of data management, and well diagrammed introduction to the subject of Data Management.

Kanter, J. (1967), "The Ubiquitous Data Base Concept", Data Processing Magazine, Vol. 9, No. 5, pp. 28-32. The author presents three problem areas that managers often overlook during implementation of a Central Data Base (CDB). The areas identified by the author are content problems, time miscalculations and file dimensions. These areas must be watched and planned more effectively by management, to receive accurate and timely information from the CDB.

Lyon, J.K. (1971), *An Introduction to Data Base Design*, Wiley-Interscience, New York, 80 pp. An extremely readable and comprehensive introduction. Organised into a logical and well-exemplified structure. Lyon writes in English, rather than Computereze. Covers all aspects of data base design on an introductory level: structure, logic, modelling, physical organisation, implementation, efficiency and examples of designs. Addressed mainly to the programmer, analyst and data processing manager.

McLaughlin, R.A. (1972), "Building a Data Base", *Datamation*, Vol. 18, No. 7, pp. 51-55. A description of assembling a data base with emphasis on problems and mistakes. Data base designed to handle the products and services of *Datamation* Magazine in order to produce a publication. A "how we did it", to allow readers to avoid the same mistakes and errors. Problems discussed are those of vendor selection, design, data collection and entry, editing and control.

Olle, W.T. (1970), "MIS: Data Bases", *Datamation*, Vol. 16, No. 15, pp. 47-50. A look at the confusion and proliferation in data base and management information systems interface with emphasis on what should be and what are the needs and requirements. An editorial-like call for organisation and standardization.

Pan, G.S. (1973), "The Characterisation of Data Management Systems", in R.A. Bassler and J.J. Logan, *The Technology of Data Base Management Systems*, Arlington: College Readings, Inc., pp. 6-8. The author describes various DBMS structuring techniques. The file organisations discussed are sequential, chaining, branch, binary, inverted, content structured and hash structures. Also discusses the computation of overhead for the different structures.

Patterson, A.C. (1972), "Data Base Hazards", *Datamation*. Vol. 18, No. 7, pp. 48-50. The author presents the problems and disadvantages of implementing a Data Base Management System from the management point of view. There are times when a data base is inappropriate: a data base is not a management information system. Article is an attempt to balance the allure of data base management systems with realities. Because data bases are the "current state of the art" thing, they may not be appropriate to a user's needs. This article asks the questions, Cost? Need? Applicability?

Patterson, A.C. (1973), "Requirements for a Generalised Data Base Management System", in R.A. Bassler and J.J. Logan, The Technology of Data Base Management Systems, Arlington: College Readings, Inc., pp. 14-16. The article distinguishes the Codasyl and Guide/Share Data Base Management System Requirements as the two main reports. He gives an excellent glossary of current jargon used in the field.

Plourde, P.J. (1973), "Data Base Dictionary Design", in Proceedings of the 1973 Cause National Conference, Boulder, Colorado: College and University Systems Exchange, pp. 699-715. This is a rather formal presentation of the subject. The author states that there is no one way to design a Data Dictionary System (DDS) because of specific local conditions. However, the design of the DDS must be the first step before the development of applications or defining the Data Base. The author provides an excellent list of data elements typically stored in dictionaries, and stresses extensive cross-referencing of data element utilisation by program, file, segment, data name, utilisation area, transaction type and table values. Data dictionaries provide three functions: describe, protect/secure and facilitate use of data in the Data Base. This article is aimed at the professional data processor, but is recommended to all serious readers.

Price, G.F. (1972), "The Ten Commandments of Data Base", Data Management, Vol. 10, No. 5, pp. 14-23. Ten "Commandments", or rules to be used when defining a data base that will help reduce chances of failure of that data base: identify data requirements; design file requirements; design system; evaluate criteria; project planning and control; data management control; develop, test and document; plan security protections; educate and implement; audit and evaluate. Excellent practical "check list" oriented to act of designing. Uses student record keeping system as example. Interesting PERT charts to explain system.

Pryrves, N.S. (1969), "Structure and Organisation of Very Large Data Bases", in F. Gruenberger, Critical Factors in Data Management, Englewood Cliffs: Prentice-Hall, pp. 127-145. The article highlights problems unique to large DBMS (Data Base Management Systems). The author stresses the need for a unified storage and retrieval subsystems, changeable file organisation (inverted files, dynamic directories), retrieval by natural language terms, and describes each of these.

Schubert, R.F. (1972), "Basic Concepts in Data Base Management Systems", Datamation, Vol. 18, No. 7, pp. 42-47. What are the basic needs of a data base management system? Article discusses concept elements, storage structure contact, logical data relationships, schema and subschema data description languages, device media control languages, and the administration of such a system. The author reviews the past failures of Data Management Systems. He states that many of today's DBMS are built from the rubble of old failures, and kept alive by vendor executives.

Scriven, D.D. (1973), "Give the Newly Appointed Some Help", in R.A. Bassler and J.J. Logan, The Technology of Data Base Management Systems. Arlington: College Readings, Inc., pp. pp. 41-42. The author suggests what information should be included in data bases. He states that research data, simulation models and numerous applications programs, as well as transaction and summary data, should be included in data bases.

Senko, M.E., E.B. Altman, M.M. Astrahan and L.H. Fehder (1972). "Data Structures and Accessing in Data-Base Systems", in IBM Systems Journal, Vol. 12, No. 1, pp. 30-105. A very thorough and rigourous discussion of an important subject, but somewhat technical for the educational manager except the first part on "Evolution of Information Systems".

Shneiderman, B. (1973), "Optimum Data Base Reorganisation Points", Communications of the ACM, Vol. 16, No. 6, pp. 362-365. The author gives techniques for timing data base reorganisation. An understanding of calculus and statistics is necessary. Recommended for the technically-oriented reader.

Shneiderman, B. (1974), "Bibliography on Data Base Structures", Data Base, Vol. 6, No. 1, pp. 3-9. Here the author has compiled the most extensive list of bibliographies on Data Base structures currently available. This is an excellent source of information for the person researching data bases.

Stross, C.O.M. (1972), "Operation of Disc Data Base", The Computer Journal, Vol. 15, No. 4, pp. 290-297. A "how we did it" rather than "how to" article. Description of Hawker Siddeley Aviation, Ltd. of England. The author describes the system and compares it in a tape vs. disc configuration. Data base organisation is discussed. Integrity is described, with two types in use at the installation: logical dum and dump. Recovery techniques for the system are described. Daily operation of the system and the reorganisation of data bases after a passage of time is discussed. An interesting description of a system.

Uhrowczik, P.P. (1973), "Data Dictionary/Directories", IBM Systems Journal, Vol. 12, No. 4, pp. 331-350. Introductory Concepts of Data Dictionary/Directory. A thorough and well organised article discussing concepts, objectives, organisation and implementation of a data dictionary/directory in a system environment. Structure diagrams are used to exemplify and amplify the written explanation. The author is an IBMer, and makes many references to the IBM literature and practices. A highly recommended article for the serious reader.

Walters, S.J. (1972). "File Design Fallacies", The Computer Journal, Vol. 15, No. 1, pp. 1-5. A refutation of the "hearsay rules of thumb". The author demonstrates that these rules, based on "hit ratio", fail so often that they should be disregarded. A reminder that file design for a complex system should not be viewed in simplistic terms, but rather in light of system design.

Wilkes, M.V. (1972), "On Preserving the Integrity of a Data Base", The Computer Journal, Vol. 15, No. 3, pp. 191-194. Once a data base is partially or completely destroyed, what then? The author discusses methods of keeping data base integrity in case of accident or error, with consideration for data structure, storage structure, and language used for interrogation and update. Security is not regarded in the problem of integrity. A discussion of copies, recovery, prevention, verification of input data, check for hardware malfunction. He also discusses a system/data management interaction to minimise loss of data integrity.

ANNOTATED BIBLIOGRAPHY TO CHAPTER 6

Churchman, W.C., R.L. Ackoff and E.L. Arnoff, (1957) Introduction to Operations Research, New York: John Wiley & Sons, Inc., Chapter XX, pp.577-594. This chapter is concerned with problems associated with gathering the evidence to test the adequacy of models and the solutions derived from them. The discussion is statistically oriented, but at a basic level. Of particular worth is the section on sampling which has the aim of using the minimum amount of time and effort in getting the information required for testing.

Irion, W.R., and A.P. Lechler, (1969) "System Validation and Verification", 14th Annual College & University Machine Records Conference, (Ann Arbor, Michigan: The University of Michigan), 11th-14th May, pp.1-7. A very readable and non-technical report on system validation and verification which are considered the first and second checkpoints, respectively, at which the system can be inspected. The end result of the validation process takes the form of a collection of documentation which provides the yardstick for verifying the system. Verification should demonstrate the ability of the system to satisfy its specifications completely and can ultimately be expanded into a completely independent audit.

Kirk, F.G., (1973) Total System Development for Information Systems (New York: John Wiley & Sons), Chapters 11 and 12. Chapter 11 is on Testing and Chapter 12 is on Cutover and Conversion. Both chapters identify the various activities involved in using a Critical Path Network Diagram.

Martin, James (1967) Design of Real Time Computer Systems, (Englewood Cliffs, New Jersey, United States: Prentice-Hall, Inc.), Chapters 37 and 38. Chapter 37 is on testing and discusses problems and techniques of testing with special reference to on-line systems. Chapter 38 is on "cutting-over" the system and examines considerations of interim operations, operator training, installation time and procedures of cut-over. Again, with special emphasis on on-line systems.

Naylor, T.H., J.L. Balinthy, D.S. Burdick, and K. Chu, (1966) Computer Simulation Techniques, (New York: John Wiley & Sons), Inc.), Chapters 8 and 9, pp.310-344. A basic knowledge of statistical methods and economic theory is a requisite to fully understanding the material presented by this distinguished group of professors. Chapter 8 explores four major methodological positions concerning the problem of verification - synthetic apriorism, ultraempiricism, positive economics and multi-stage verification. The design of simulation experiments is treated in Chapter 9 with the purpose of relating the problem of designing computer simulation experiments to the literature on experimental design techniques.

SELECTED ANNOTATED BIBLIOGRAPHY TO CHAPTER 7

Chapin, N., (1963) *An Introduction to Automatic Computers* (2nd ed.), (Princeton, N.J.: D.Van Nostrand Company, Inc.) pp. 388-93 Maintenance has the objective of keeping the system applications performing adequately; it applies to equipment, the applications programs and procedures. Two major types of maintenance are preventive maintenance and corrective maintenance. Preventive maintenance attempts to detect and correct the probable causes of failure before they actually bring about a malfunction. Corrective maintenance attempts to detect and correct the cause of malfunction after the malfunction has come about. Changes in applications may call for redoing part of the system analysis, systems design, programming, debugging and conversion. The author points out that it has been found unwise to attempt to make changes each time requirements change - it is wise to make a group at a time and only after the changes have been authorised.

Couger, J.D. and R.W. Knapp, (1974) *Systems Analysis Techniques*, (New York: John Wiley and Sons). A set of readings with an excellent selection of three articles on Cost/Effectiveness Analysis that can be used in evaluating information systems. Two articles highly recommended are "Cost/Benefit Analysis of Information Systems" by Emery and the "Economic Evaluation of Management Information Systems " by Boyd and Krasnow. Both articles are oriented towards the business view but deal with principles that can be easily transferred to an educational setting. Both articles are not mathematical though Emery makes extensive use of curves in discussing the value and cost of differing quantities of information.

Davis, Gordon B., (1974) *Management Information Systems*: *Conceptual, Foundations, Structure and Development*, (New York: McGraw Hill Book Company), Ch. 16. An excellent discussion of evaluation of Information Systems including that of hardware/software systems, applications. Also discusses cost/benefit analysis of applications and design alternatives.

Kirk, F.G., (1973) *Total System Development for Information Systems*, (New York: John Wiley and Sons) Ch.14. Discusses both the evaluation and maintenance phases of development. Uses the Critical Path Diagram to identify and discuss the different activities involved. Easy reading.

McDonough, A.M., and G. Garrett, (1965) Management Systems. (Homewood, Ill.: Richard D. Irwin, Inc.) pp. 184-96. The whole tenor of this book is that systems do not remain fixed and that changes will be made. An understanding of the material on feedback and evaluation of systems effectiveness presented by the authors is requisite to developing a full appreciation of maintaining a viable system. In the context of the discussion presented, feedback can be considered as a synonym for the planning and control cycle in management. The key element in the feedback process is presented as the criteria element, as it is against the manager's criteria that the evaluation of system effectiveness will be made.

Nolan, R.L. and H. H. Seward, (1974) "Measuring User Satisfaction to Evaluate Information Systems," in Nolan, R.L. (ed.), Managing the Data Resource Function (New York: West Publishing Company), pp. 253-276. Describes the use of a questionnaire in determining user satisfaction of information system.

Riggs, R, (1969) "Computer Systems Maintenance," Datamation Vol. 14, No. 11, pp 227, 231-2. Describes three major steps in establishing a systems maintenance function "(1) completely document all systems and programs currently in production, (2) establish a log of maintenance activity, and (3) organise" for maintenance.

Rothary, B., (1968) Installing and Managing a Computer, (New York: Brandon/Systems Press) pp. 135-42. This presents a basic treatment on system maintenance. The author conceptualises change as one of the greatest forces in the world today, e.g., changes in requirements, new procedures to be inserted, old procedures dropped, etc. Two dangers that arise from change are: (1) Redundant procedures may be left in the programs, branched around or by-passed, and (2) the changed routines may create an even bigger split between the working program and its original description or specification.

ANNOTATED BIBLIOGRAPHY TO CHAPTER [illegible]

Allen, B. (1968) "Danger Ahead! Safeguard Your Computer". Harvard *Business Review*, XLVI, No.6, pp.97-101.
This article includes case studies of disastrous damage suffered by processing centres. It tells how magnets were used to destroy all of one company's data files and programs. It also describes the collapse of a centre on the second floor caused by a fire on the first floor. The need for back-up equipment is discussed.

Burch, J.G. Jr., and F.R. Strater Jr., (1974) *Information Systems: Theory and Practice*. Santa Barbara, California. Hamilton Publishing Company, pp.306-324.
An excellent chapter on control considerations for the design of an information system. A text used in computer science courses in information systems.

Davis, G.B. (1968) *Auditing & EDP*; New York: American Institute of Certified Public Accounts.
This very informative, straightforward book was prepared by a task force of AICPA members directed by Professor Davis, on leave from the University of Minnesota as consultant to the AICPA. Chapter IV describes equipment malfunctions in general and the auditing controls that are associated with major items of equipment. Chapter V discusses control over input and output, and emphasizes that system design should include provisions to ensure accuracy and completeness. Chapter VI stresses the need for proper organisation and supervision of programming. Chapter VII surveys the problems of safeguarding records and files; control measures such as physical standards, procedural controls, retention plan, recovery plan, and insurance are described. In Chapter VIII, principles of internal control are reviewed in relation to computer-based systems; a questionnaire for evaluating internal control in EDP is discussed. The effects of EDP on the audit trail are discussed in Chapter XIX, which includes a summary of guidelines for design of the audit trail.

"Auditing Fast Response Systems" *EDP Analyzer*, Vol.5, No.6, (1967) pp.1-12. This article details the important changes in the auditing environment brought about by increased use of fast-response computer-based systems. Changes in the visible audit trail, different chances for accidental or deliberate damage to date, increased complexity of operation, and changes in people problems have complicated the conduct of the audit. It is suggested that good management will require auditing "through" the computer rather than "around" the computer.

Harrison, John P. (1968) "An Auditor's View of Data Processing". *Journal of Data Management*, September, pp.32-36.
The audit trail is defined as the capacity to trace any transaction from its point of origin to its final place in the accounts or to trace back from its end result to its origin. Although there are few long-standing practices applicable to auditing EDP, the author does list and discuss some preferred practices for control in the areas of organisation, input, hardware, software, programs, output, and systems.

Hearn, C.P. (1965) "TACT Will Improve Your Input Data". *Datamation*, December, pp.58, 63-64.
The greatest challenge in the science of data processing is that of consistently collecting input data that is timely, accurate, and complete. One method suggested as an answer to this challenge involves "TACT", or "Timeliness, Accuracy and Completeness Techniques". In general, the TACT method requires that all input data be edited for accuracy, timeliness, and completeness, and that reports of errors be fed back to the originator.

Kindred, A.R. (1973) *Data Systems and Management*: *An Introduction to Systems Analysis and Design*. Englewood Cliffs, New Jersey: Prentice Hall, Inc., pp.247-260.
Has an excellent chapter on "Verification, Control and Auditing". The chapter also has a bibliography with excellent and extensive suggestions for further reading, especially good on auditing.

Management Control of Electronic Data Processing. White Plains, New York: International Business Machines Corporation, 1965.
This manual describes some methods for controlling data-processing procedures from the standpoint of management. It includes a very complete and useful checklist for reviewing the adequacy of controls; it also gives two good examples of effective controls in an EDP system.

Miccio, J.V. (1967) "Electronic Data Processing: Its Controls and Economics". Journal of Data Management, June, pp.20-25.
Dr. Miccio defines the major objective of control as insuring that only valid data is accepted and processed, and that no invalidity is introduced during processing. Several methods of achieving the objective are treated: editing of input data, use of check digits, checking for logical relationships, proving program reliability, use of tape labels, use of redundant identifiers, control of tape libraries, etc. Also included is a brief, but very worthwhile discussion of the cost aspects of control and the economics of EDP.

McRae, T.W. (1964) The Impact of Computers on Accounting. London: John Wiley & Sons Ltd.
In this discussion of auditing and control, emphasis is given to the importance of checking the system rather than vouching the individual items of data processed by the system. The author suggests and discusses four main methods of determining if the system is running according to plan: the "black box" approach; the job simulation method, the audit-in-depth approach, using statistical sampling; and the method of auditing the job file directly. Automatic checking by machine, programmed checking, and controls applied to magnetic tape are introduced and treated in a general fashion. The material will be of interest to those with little previous knowledge of electronic data processing.

Newman, M.S. (1964) "Internal Control and Data Processing". Financial Executive, November, pp.45-52.
The best foundation for sound internal control in a data processing installation is the documentation of the systems, programs, procedures, operating instructions, and tests. Mr. Newman discusses the documentation requirements in terms of: (1) a comprehensive flowchart giving an overall view of the workings of the system; (2) specific flowcharts for details of the system; and (3) diagrams for details of programs. Other matters related to control of data processing are discussed: organisation and assignment of duties, programmed and built-in controls, control over source documents, and control over output.

Porter, W.R.,Jr. (1965) "A Control Framework for Electronic Systems". The Journal of Accountancy, October, pp.56-63.
This article discusses the types of controls in an electronic data-processing system, and their relationship to the design and implementation of a management information system. The administrative controls are discussed under the three broad

areas of systems design, programming, and computer operation. Emphasis is placed on the need for complete documentation in each of the areas. Procedural controls over source data, processing, and output also receive attention. The author concludes by stating that the design of a management information system dictates spending substantial amounts of time and effort in the formulation of controls.

Turner, T.E. (1969) "An Economic Look at Quality". Quality Progress, April, pp.15-21.
Although the article is intended for the quality-control professional, it will be meaningful to any reader with a knowledge of basic micro-economics. The analysis of costs versus degree of control has application in the matter of quality control in EDP. The material is well illustrated with appropriate analogue models of the costs of inspection required to maintain the specified degree of quality.

ANNOTATED BIBLIOGRAPHY TO CHAPTER 9

Mosmann, C. and J. Caffrey, Computers on Campus, Washington, D.C. American Council on Education 1967. One of the first and best books on the role and status of computers on university campus. Discusses their extent, applications and administration.

Cahill, J.J. (1970) "A Dictionary/Directory Method for Building Common MIS Data Base", Journal of Systems Management, Vol.21, No.11, pp.1-14. Defines the characteristics of a common data base. Encompasses the definition of guidelines for establishing information for a common data base, data specifications and specification statements for data, use of the dictionary/directory, and also through the maintenance and updating of the dictionary/directory. (The article is derived primarily from experience and methods developed by IBM).

Canning, R.G. (1972) "The 'Data Administrator' Function", EDP Analyser, Vol.10, No.11, pp.1-14. A new job function having the title of "data administrator" has, in recent times, emerged in many major United States and world companies. With this emerging function has come a strong need for defining the boundaries and guidelines of the position. This paper provides a look at many such positions as they exist in case studies of certain United States companies, and along with these studies are excerpts from works by different committees, groups and organisations defining the actual functions of these data administrators. The article contains a very comprehensive set of guidelines and job definitions/descriptions for the data administrator, monitoring function as well as design and operation activities. Specific examples are given for the Pillsbury Company, Wellsco Data Corporation and Pacific Gas and Electric, as well as functions of the DBA by Codasyl DBTG, Guide-Share DBRG and others.

Cuozzo, D.E. and J.F Kurtz, (1973) "Building a Base for Data Base: A Management Perspective", Datamation, Vol.19, No.10, pp.71-75. The purpose of this article is to help potential users make a Data Base Management System go or no-go decision. The authors present key features, benefits, and disadvantages of implementing a DBMS, and advise managers to plan thoroughly and prepare for future implementation of the system. It examines some of the serious considerations a MIS/EDP manager must undertake concerning the one-time and ongoing incremental cost of a Data Base Management System (DBMS). It also provides significant insight into the reduction of cost by initiating planning and preparation programs for both future data base facilities and possible short-term alternatives.

Goldstein, R.C. (1974) "Tutorial on Computer Data Bases", in Nolan, R.I. Managing the Data Resource Function, New York: West Publishing Company, pp.349-389. This article examines five main areas of concern pertaining to this subject. The role of the data base in the organisation, technical aspects of organising information, interfacing between the data base and users, and privacy and security are discussed in some detail. Also discussed are considerations of file design, file structure, data base organisation and addressing techniques, and access mechanisms. It is also good on physical data structures. An excellent tutorial.

IBM, Management Planning Guide for a Manual of Data Processing Standards not dated, Form C20-1670-x. This manual, regularly updated, is an excellent document on the management and administration of a computing centre. It discusses standards to be adopted and enforced necessary for an efficient and effective computing centre.

Mosmann, C., Academic Computers in Service, San Francisco, Jossey-Bass Mc. 1973. This is a sequel to the book by Caffrey and Mosmann. Discusses computers in educational institutions in the U.S.

Timmerman, J. (1972) "Managing the Data Base", Data Processing Magazine, Vol.14, No.1, pp.7-8. The author explains the importance of an integrated company-wide data base, and points out nine essential responsibilities of the data base manager.

Traver, C.M. (1973) "Data Base Administrator", paper presented at the College and University Machine Records Conference, 10 pp. The author was a Data Administrator at Stanford University when the OASIS System was developed and used. Many experiences of the newly-evolving role are discussed. Functions of the Data Administrator and his needed qualifications are discussed.

A GLOSSARY IN PROSE
ON
INFORMATION SYSTEMS(1)

by

K.M. Hussain

1. INTRODUCTION

This is an attempt to introduce information systems terminology in a meaningful context(2). The narration will also serve the purpose of giving a partial preview and summary of the IMHE Programme seminar on "Information Systems for Institutions of Higher Education".

2. INFORMATION SYSTEMS

Most educational institutions have information systems whose main function is to provide information when it is needed, where it is needed, and in the form in which it is needed. To achieve this, the development of an information system must be carefully planned and executed. This development is performed in stages and consists of a set of jobs called activities.

The first group of activities, or stage in the development process, is a feasibility study which examines alternatives for practicality within organisational constraints. A constraint is a force that places a limit on what is possible.

Once an alternative is chosen as being acceptable to the administrator, the user of the system, the next stage in the development process is for him to define his needs more specifically. To achieve these needs, a system is designed and then implemented.

There are many types of information systems. The earliest were manual systems which were followed by electrical accounting machines, more commonly referred to as EAM equipment. These were superseded

---

1) The following has been extracted from Development of Information Systems for Education by Khateeb M. Hussain, (c) 1973. It is reprinted by permission of Prentice Hall, Inc., Englewood Cliffs, New Jersey, United States.

2) For a prose glossary on computers that is more technical than this one (and highly recommended) see J. Caffery and C.J. Mosmann, (1967). Computers on Campus, Washington, D.C., American Council on Education, pp.187-193.

by *computers*, which are electronic machines capable of performing complex operations with data. These operations are performed very quickly, accurately, and automatically. But computers must be instructed precisely as to what operations are to be performed and the sequence in which they are to be performed. These sets of instructions are called *programs*.

There are many *programming languages* in which programs are written. One group of such languages is called *machine languages*, also referred to as *lower-level languages*. The programs in machine language are a set of instructions stated in coded numbers that are directly recognised by the machine. Other programming languages are more like the English language and are referred to as *higher-level languages*. These are more independent of a particular machine than are machine languages. Their usage often reduces the conversion costs that result from changing from one computer to another. An example of such a higher-level language and one most commonly used for information systems and business data processing is *COBOL*. Languages used mainly for scientific programming are *ALGOL*, *MAD*, *APL*, and many others. There are some languages that are often used for both scientific and business data processing. Examples of these languages are FORTRAN and PL/1. Programming languages are used to communicate with a computer and are distinct from languages used to communicate between people, which are known as *natural languages*.

This paper is concerned largely with computer-based information systems. It will not discuss systems that are entirely EAM oriented, but will be concerned with EAM equipment inasmuch as it is used to support computer processing. Among such equipment is the *keypunch* which punches holes in a *data card* to correspond to data according to a specific coding system; a *verifier* that identifies errors in keypunching; a *reproducer* that reproduces cards with possible variations in the locations of specific sets of holes in them; a *sorter* that classifies cards according to desired data classifications; and a *collator* which merges data cards - that is, combines two similarly sequenced sets of cards into one set. These devices are concerned with preparing data that goes into the computer and are, therefore, referred to as *input equipment*. Devices that handle data coming out of the computer are called *output equipment* and include a *printer* that prints output; a *decollator* that separates the carbon sheets from multiple sheets of paper, called a multiple *ply* paper; and a *burster* that breaks the perforations between sheets in order to separate them. These devices, except the printer, are also referred to as *off-line* devices because they are not connected to the CPU (Central Processing Unit), the unit that performs the computational and control functions in a computer.

When devices are in direct communication with the CPU, they are referred to as *on-line* equipment. Examples are printers and terminals that receive input or produce output directly from the CPU. These may be *typewriter terminals* or *CRTs* (Cathode Ray Tubes). A *CRT Terminal* looks much like a television set sitting on top of a typewriter. The terminal may be physically part of the computer, in which case it is called a *console*. It can also be located remote from the computer, in which case it is said to have *remote access*; it is connected to the computer by direct cable or by telephone. In cases of telephone transmission, there is a need for converting voice signals of the telephone into data acceptable for the terminals. This is done by a *data-phone* with a *data transmission set*. The data-phone in this case performs the function of an *interface* between the terminal and the telephone lines. Interfaces are also required between the computer and its *on-line peripheral devices*. Examples of on-line peripheral devices are a printer that prints output, a *plotter* that plots output graphically, a *card reader* that reads punched cards, and an *optical scanner* or *optical reader* that recognises marks and special characters on documents. The characters can be of different styles and sizes, referred to as *fonts*.

The equipment discussed above is called *hardware*. Different combinations of equipment are referred to as *configurations* of equipment. The selection of the best equipment configurations is part of the system design stage in the systems development process.

In contrast to hardware, there is *software*, which is the set of computer programs. A program instructs the computer on the *algorithm* to be used, that is, the specific computing procedure to be followed in order to achieve the desired output. Such a program is sometimes called an *applications program*. In contrast there is a set of programs that translates and interprets the applications programs from a higher-level language into machine language. These are called *compilers*, *assemblers*, *translators*, and *interpreters*. Other computer programs govern the scheduling of *jobs* (user's programs that have been processed by the computer), and automate the relationship of the computer and its peripheral devices. These are called *monitors* or *supervisors*. Still other computer programs perform "household" duties of automating the operations of the computer and performing frequently required functions, such as *listing* an information file. These are called *utility* programs.

The compilers, assemblers, translators, interpreters, monitors, supervisors, and utility programs are collectively referred to as *systems programs* and are frequently provided with computer equipment by the manufacturers. These programs are distinct from *applications* programs that are typically provided by the user. Collectively they constitute what is known as *software*.

Systems programs and techniques are referred to as operating systems. The operating system along with the hardware configuration is unique for most computer systems. This is why programs run on one system are not easily run on another. If two computer systems can run the same set of computer programs they are considered compatible with each other. One system can then serve as a backup for the other in the event that one breaks down.

This paper is addressed to the administrator in education. Typically, he will not be concerned with systems programs and techniques related to hardware or operating systems. He is, however, interested in the successful operations of the information system. To achieve this he must participate actively in the design of the output, which is the result he will receive, as well as the input, which is the data and resources that are put into the system. The hardware and software of the system should then produce the desired results. The interrelationship between man and the machine system (hardware and software) is governed by a set of instructions referred to as procedures.

Part of system design, especially of an information system, is concerned with the structuring and storing of data so that it can be efficiently and effectively processed. This is known as data management or file management. It is concerned with a data base, also called data bank, which consists of a set of integrated files. A file is a set of records; a record is a set of data elements; a data element is something with a value worth recording. Each data element is represented by a data field, which is a set of data columns. On a data card there are 80 data columns, each representing a character of data. Data is the record of a fact or event which, when selected and processed (individually or with other data), is called information.

Input data and programs to be processed are stored on a storage or memory device.

There are many types of memory devices. One is called core. It is part of the computer equipment and is referred to as primary storage or internal storage. An external memory device example would be a magnetic tape similar to that used for recording songs. It is more suitable for recording data that is to be processed and retrieved sequentially. An example would be a payroll or student grade processing at the end of a semester. If the processing is to be done on an individual basis, data must be processed and retrieved in random order. The memory device appropriate for such random processing is a disk which is similar to a phonograph record. The tape (magnetic or paper tape) and the disk are referred to as auxiliary, secondary, or external storage. They supplement the internal core storage.

Most data changes. This change is recorded as a transaction. A file containing all transactions is called a transaction file or detail file. It changes the original data on the master file. The reflection of new transactions is referred to as updating, and periodic updating of a master file is referred to as file processing. The deletion of irrelevant data is referred to as purging of data.

After the input, output, files, and procedures are designed, the design specifications are implemented and the new system is then tested. This consists of a comparison of existing performance with desired performance. Often they do not match. Problems may exist with the computer programs. These are detected, located, and corrected, which is referred to as debugging.

Performance is again compared with desired performance. Once the system appears to perform as expected, it is then documented, which is the process of stating all relevant facts about the system. This documentation includes: flowcharts and decision tables, which specify the logic of the decision rules and show the flow of data and information. The documentation and programs are then deposited in a program library and are handled, maintained, and controlled by personnel referred to as librarians.

After successful testing and documentation, the system is implemented and then evaluated. The evaluation should be based on both efficiency and effectiveness. Efficiency refers to the relationship of output and input. Effectiveness refers to the successful achievement of critical factors of performance set by the user. Examples of critical factors are accuracy, which is freedom from error, and timeliness which concerns the time availability of information.

Computer operations have many modes of operation. One is batch processing which involves the collecting of jobs into a batch before they are processed. Another is on-line time-sharing where the user is serviced when his turn comes up. However, because of very fast processing speeds the user is serviced almost instantly and has the illusion of having the machine all to himself and dedicated to his use. Many educational institutions have time-sharing systems, if not for administrative purpose, then for educational services.

Some educational institutions allow their students to run their computers. This mode of processing is called open-shop, in contrast to the closed-shop situations where the computer room is closed to all but professional personnel responsible for operations, known as computer operators.

There are many other professional personnel involved in the development of systems. The most important of these is the systems analyst. In developing a complex system, there is often a team of systems analysts also referred to as systems designers and systems engineers. Among them, the team members have an expertise in analysing and synthesizing systems; in Operations Research and Management Science, which is the use of mathematical and statistical techniques for finding the optimal (best) solution; in industrial engineering; in applications programming; and in data management. The team of systems analysts also has some knowledge of computer hardware and operating systems, and calls on hardware and systems programming specialists for help when needed.

The systems analyst sometimes does his own computer programming and sometimes he assigns this responsibility to a programmer, who is a professional at writing programs. If the logic and flow of the program is specified in detail, then the actual writing of the program is done by a lower-level programmer, called a programming coder.

Another person referred to as a coder is a clerk who represents lengthy data by abbreviated codes. The codes may be alphabetic, numeric, or a combination of alphabetic and numeric symbols called alphanumeric or alphameric.

The use of coding to reduce the size of data is one of many types of data compaction techniques. The data, after being recorded in written or coded form, is then converted to symbols that can be read by a machine. Data once converted in this way is referred to as being machine-readable. An example is a data card with punched holes, also referred to as a Hollerith card named after the person who invented data cards in their present form in 1889.

If data conversion is done on a keypunch, then it is done by a keypuncher. The cards are sometimes also processed on EAM equipment by a TAB operator. Other professional and technical personnel involved in systems work are the systems programmer, who writes systems programs, and the control clerk, who checks and controls the quality of output. The checking should be done after each run which is a computational attempt with a set of data. If the output meets the prescribed standards, it is distributed to authorised personnel.

The run-time may be a few seconds or a few minutes. But developing a system that will produce a desired run of a few seconds on a computer often takes many days, months, or even years.

With modern computers, the run-time is very small compared to the response-time, which is the time elapsed from the moment the input starts being read to the moment the output is produced. It includes the time required to wait before the computer Central Processing Unit is available, and the time for reading and printing. In information systems, the time spent in input-output operations is typically very large compared to the time spent on computations, and hence it delays the operations. Such a computer system is referred to as being input-output bound, as opposed to scientific machines that are limited on computing capabilities and are referred to as being compute bound.

In addition to the response-time, there is also time spent in preparing input and time spent in the control of output and its distribution. These are, relatively speaking, the more time-consuming and error-prone operations, since they are very _people intensive_: that is, these jobs are mostly done manually by people.

One other time concept concerns _lapse time_. It is the time that it takes from the start of a job to the finish, and not necessarily the time actually spent on the job. For example, an administrator may take ten minutes to make a decision, but it may take him three weeks of preparation to make it. The lapse time is then three weeks and ten minutes. Similarly, the lapse time for preparing computer programs is much longer than the actual time spent in _writing_ a program, since it includes the hours often spent in waiting to use the computer for debugging. These long lapse times for administrative decision-making and program preparation are the reasons why the development of information systems takes so long and why they must be planned with adequate _lead-time_.

Systems personnel, user personnel, hardware, software, procedures, and input data constitute the basic components of a system. The resources are organised in to specific jobs and activities, the collection of activities being known as a _project_. The activities of the project must be _sequenced_ and _scheduled_ and these must be controlled for the timely completion of the project. This is done by a _project leader_ or _project manager_ who is responsible for the planning, organising, and implementation of a systems project.